Networking protocols and administration

ITV8030

lecture 6

multicast, dhcp, dns

lecture plan

- **Broadcast and multicast : very brief intro**
  - Sending ip packets to several hosts at once
  - Broadcast f.ex arp, dhcp requests etc
  - Multicast f.ex television
- **Dhcp**
  - Setting host IP address, gateway, dns server automatically on the local network
- **DNS**
  - Finding ip addresses for domain names like www.ttu.ee
- **Summary example**
  - Ordinary http request from the browser: what happens

- using mostly slides from Univ of Virginia / Univ of Toronto, plus texts from other sources

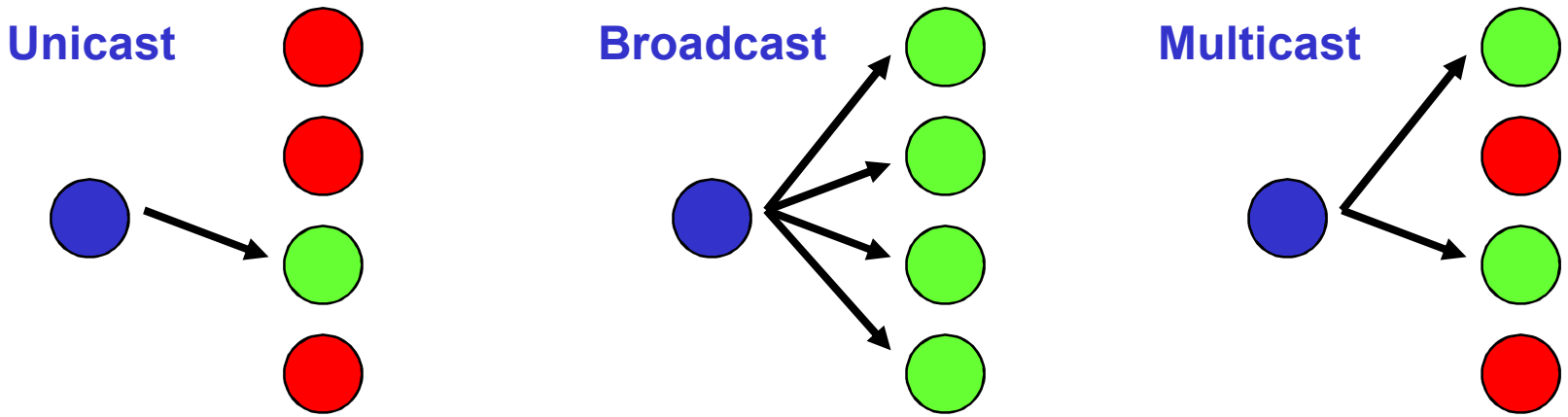
Broadcast and multicast

IP Multicast model:

“You put packets in at one end, and the network conspires to deliver them to anyone who asks.”

Broadcasting and Multicasting

- Multicast communications refers to one-to-many or many-to-many communications.



Ethernet broadcast example

- ARP query: who on this network has IP address X.X.X.X? Please send me your MAC address.
- Query sent to the special MAC address

FF-FF-FF-FF-FF-FF

- Every host on the local network will listen.

Internet broadcast addresses

- Special addresses not routed outside local network.
- Set all host number bits to ones.
- For class C addresses (last 8 bits are host address) the last byte will be 255:
  - X.X.X.255

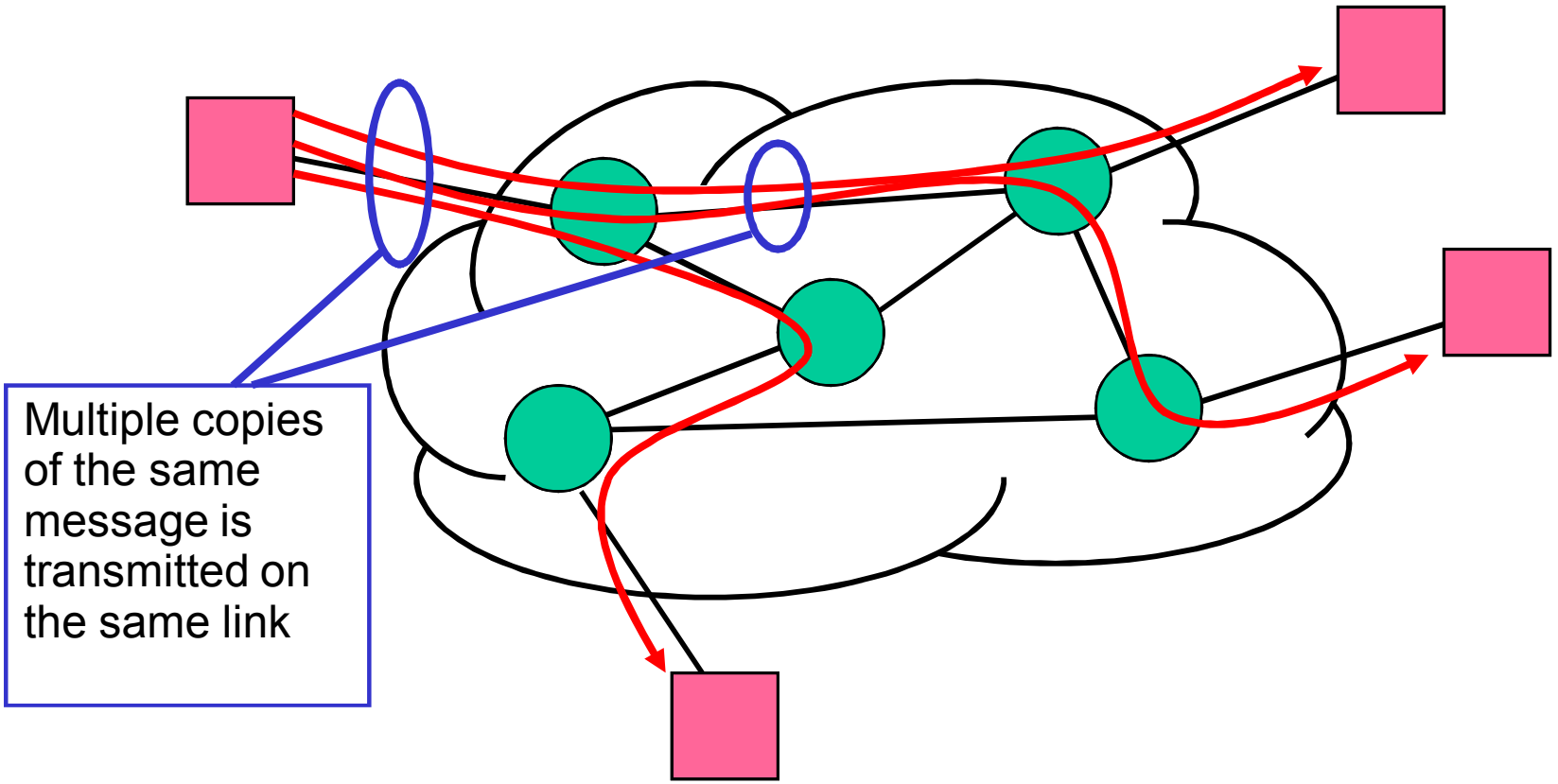
- Used f.ex for RIP configuration protocol advertisements and DHCP protocol queries:
  - Whenever we do not know the addressee IP
  - as said before, broadcast packets do not exit local network!

Applications with multiple receivers

- Many applications transmit the same data at one time to multiple receivers
    - Broadcasts of Radio or Video
    - Videoconferencing
    - Shared Applications
- A network must have mechanisms to support such applications in an efficient manner

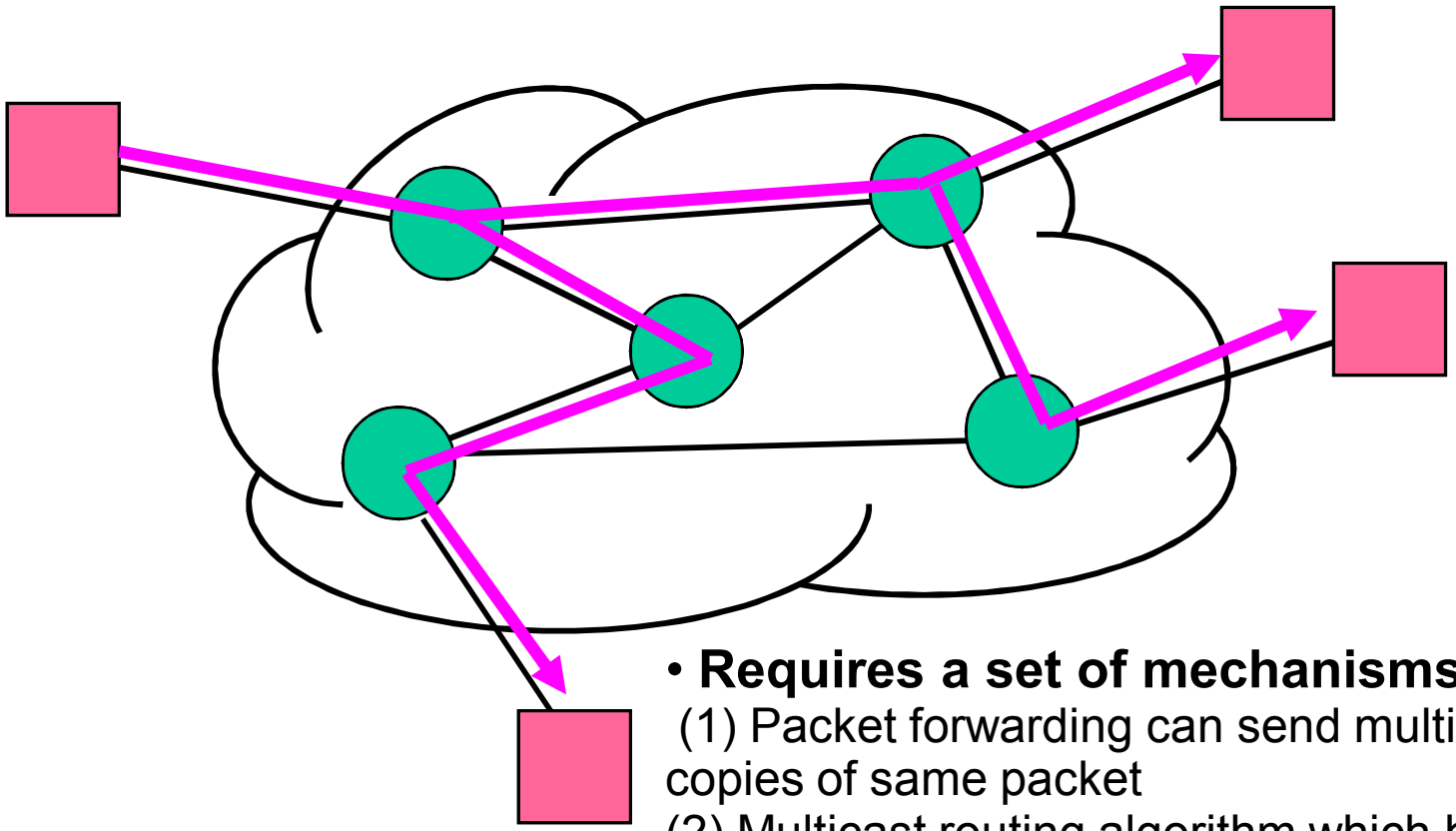
Sending same data point-to-point:

•



Multicasting same data

•



• **Requires a set of mechanisms:**
(1) Packet forwarding can send multiple copies of same packet
(2) Multicast routing algorithm which builds a spanning tree (dynamically)

Multicast Groups

- The set of receivers for a multicast transmission is called a **multicast group**
  - A multicast group is identified by a **multicast address**
  - A user that wants to receive multicast transmissions **joins** the corresponding multicast group, and becomes a **member** of that group

- After a user joins, the network builds the necessary routing paths so that the user receives the data sent to the multicast group

Prime example: telco IPTV (elion...)

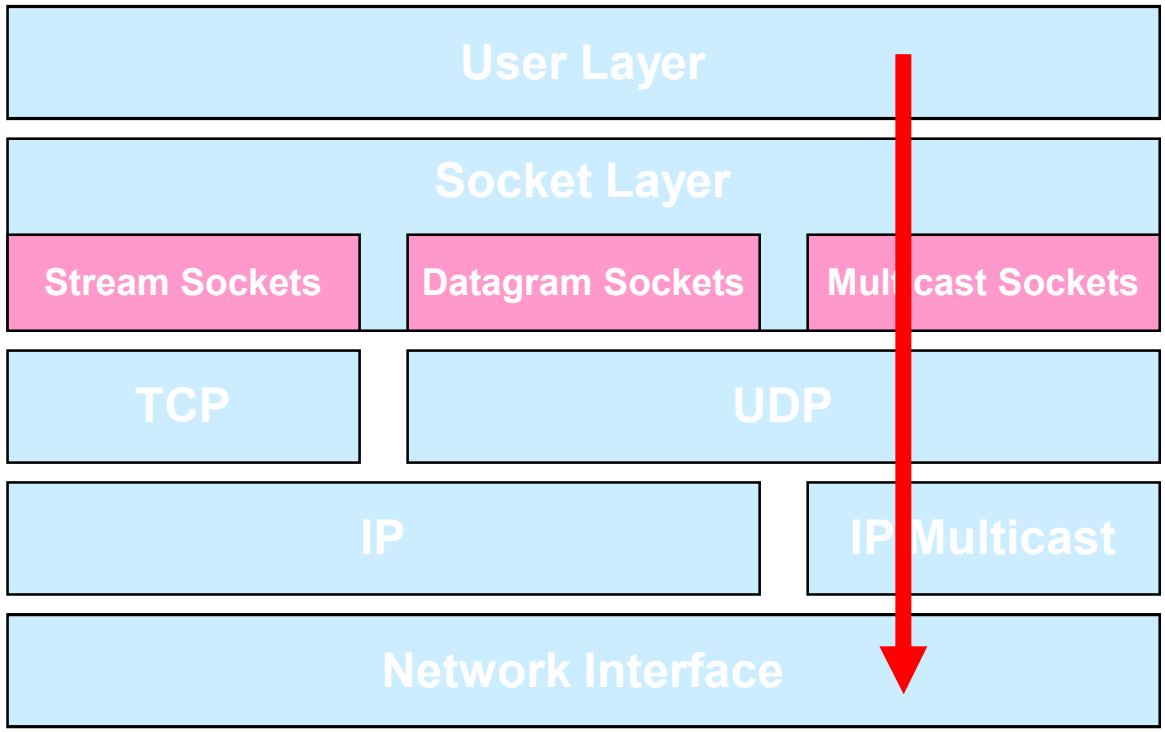
- How do you send hundreds of channels out to an IPTV subscriber with a DSL line?
- Simple: you only send a few at a time. When a user changes the channel on their set-top box, the box does not "tune" a channel like a cable system.
- There is in fact no such thing as "tuning" anymore—the box is simply an IP receiver. What happens instead is that the box switches channels by using the IP Group Membership Protocol (IGMP) v2 to join a new multicast group.
- When the local office receives this request, it checks to make sure that the user is authorized to view the new channel, then directs the routers in the local office to add that particular user to the channel's distribution list.
- In this way, only signals that are currently being watched are actually being sent from the local office to the DSLAM and on to the user.

Semantics of general IP Multicast

- Multicast groups are identified by IP addresses in the range 224.0.0.0 - 239.255.255.255 (class D address)
- Every host (*more precisely:* interface) can join and leave a multicast group dynamically
    - no access control
- Every IP datagram send to a multicast group is transmitted to all members of the group
    - no security, no “floor control”
    - Sender does not need to be a member of the group
- The IP Multicast service is unreliable

The IP Protocol Stack

- IP Multicasting only supports UDP as higher layer
- There is no multicast TCP !



IP Multicasting

- There are three essential components of the IP Multicast service:

IP Multicast Addressing
IP Group Management
Multicast Routing

Multicast Addressing

- All Class D addresses are multicast addresses:

Class D

Class	From	To
D	224.0.0.0	239.255.255.255

- Multicast addresses are dynamically assigned.
- An IP datagram sent to a multicast address is forwarded to everyone who has joined the multicast group
- If an application is terminated, the multicast address is (implicitly) released.

Types of Multicast addresses

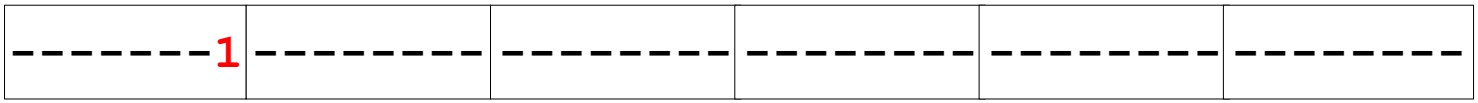
- The range of addresses between 224.0.0.0 and 224.0.0.255, inclusive, is reserved for the use of routing protocols and other low-level topology discovery or maintenance protocols
- Multicast routers should not forward any multicast datagram with destination addresses in this range.

- Examples of special and reserved Class D addresses, e.g,

224.0.0.1	All systems on this subnet
224.0.0.2	All routers on this subnet
224.0.1.1	NTP (Network Time Protocol)
224.0.0.9	RIP-2 (a routing protocol)

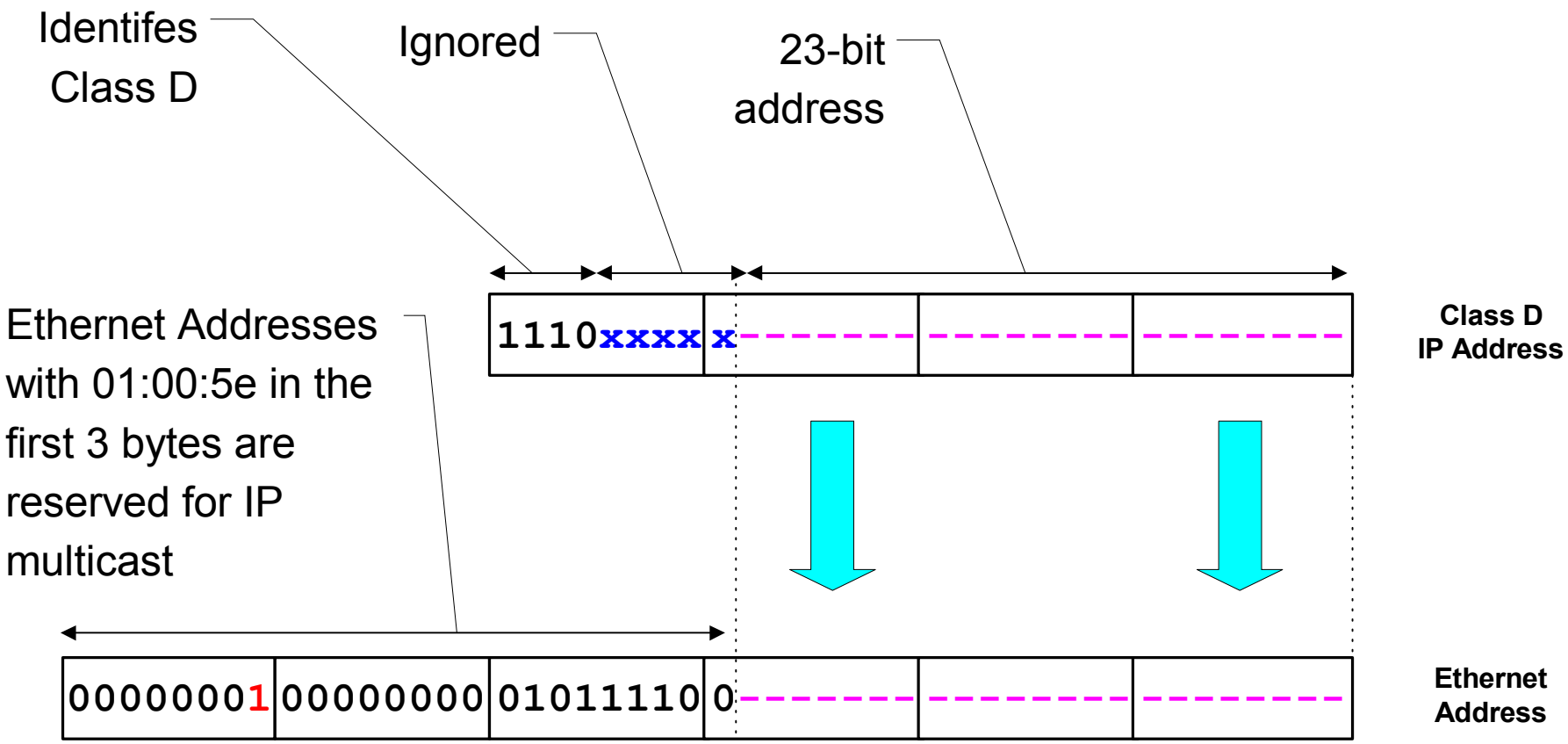
Multicast Address Translation

- In Ethernet MAC addresses, a multicast address is identified by setting the lowest bit of the “most left byte”



Not all Ethernet cards can filter multicast addresses in hardware
Then: Filtering is done in software by device driver.

Multicast Address Mapping



IGMP

- The **Internet Group Management Protocol (IGMP)** is a simple protocol for the support of IP multicast.
- IGMP is defined in RFC 1112.
- IGMP operates on a physical network (e.g., single Ethernet Segment.
- IGMP is used by multicast routers to keep track of membership in a multicast group.
- Support for:
  - Joining a multicast group
  - Query membership
  - Send membership reports

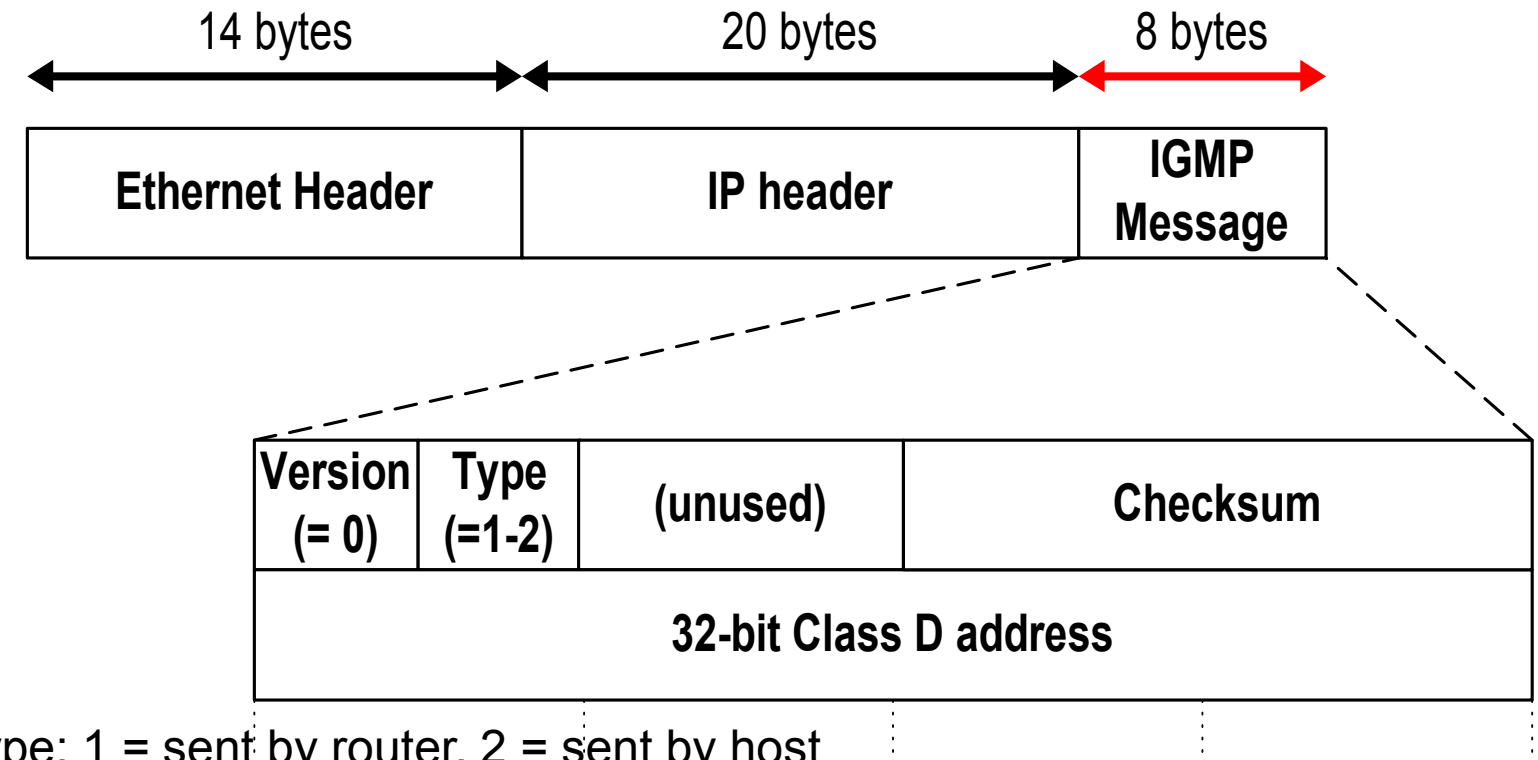
IGMP Protocol

- A host sends an IGMP report when it joins a multicast group (Note: multiple processes on a host can join. A report is sent only for the first process).
- No report is sent when a process leaves a group
- A multicast router regularly multicasts an IGMP query to all hosts (group address is set to zero).
- A host responds to an IGMP query with an IGMP report.

- Multicast router keeps a table on the multicast groups that have joined hosts. The router only forwards a packet, if there is a host still joined.
- Note: Router does not keep track which host is joined.

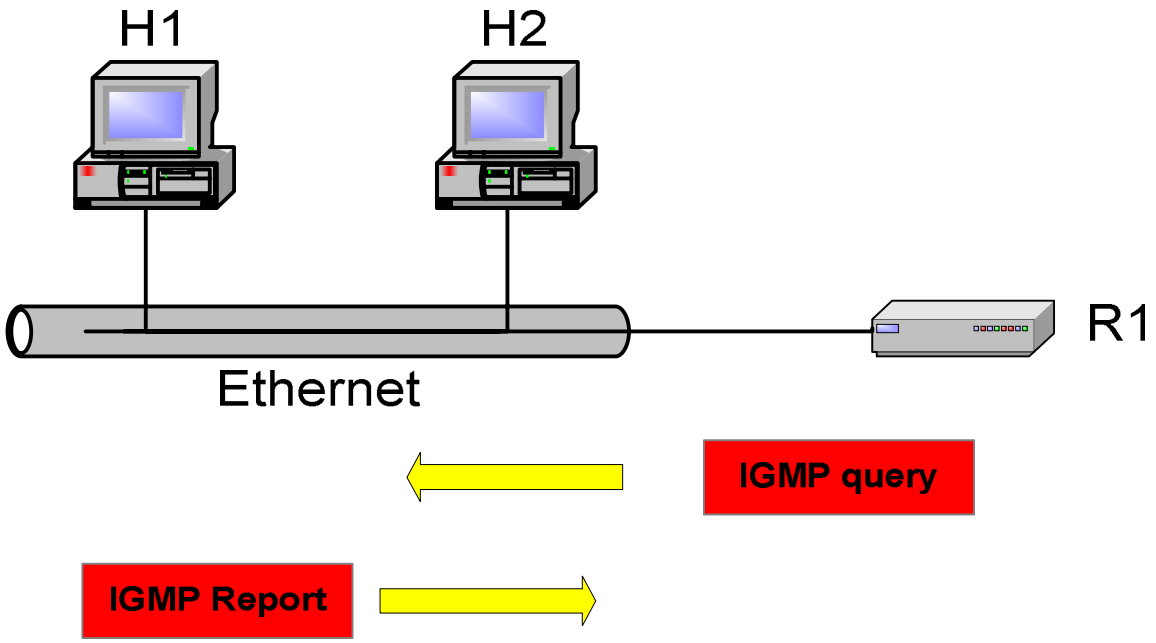
IGMP Packet Format

- IGMP messages are only 8 bytes long

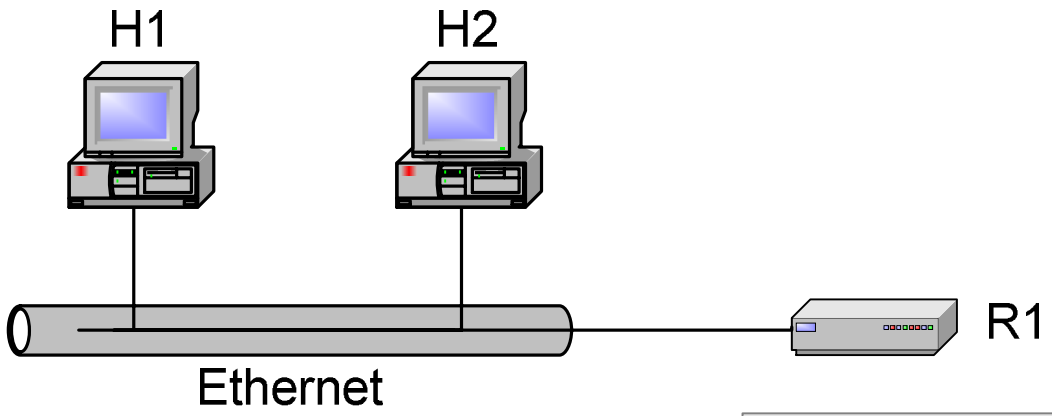


•Type: 1 = sent by router, 2 = sent by host

IGMP Protocol



IGMP Protocol



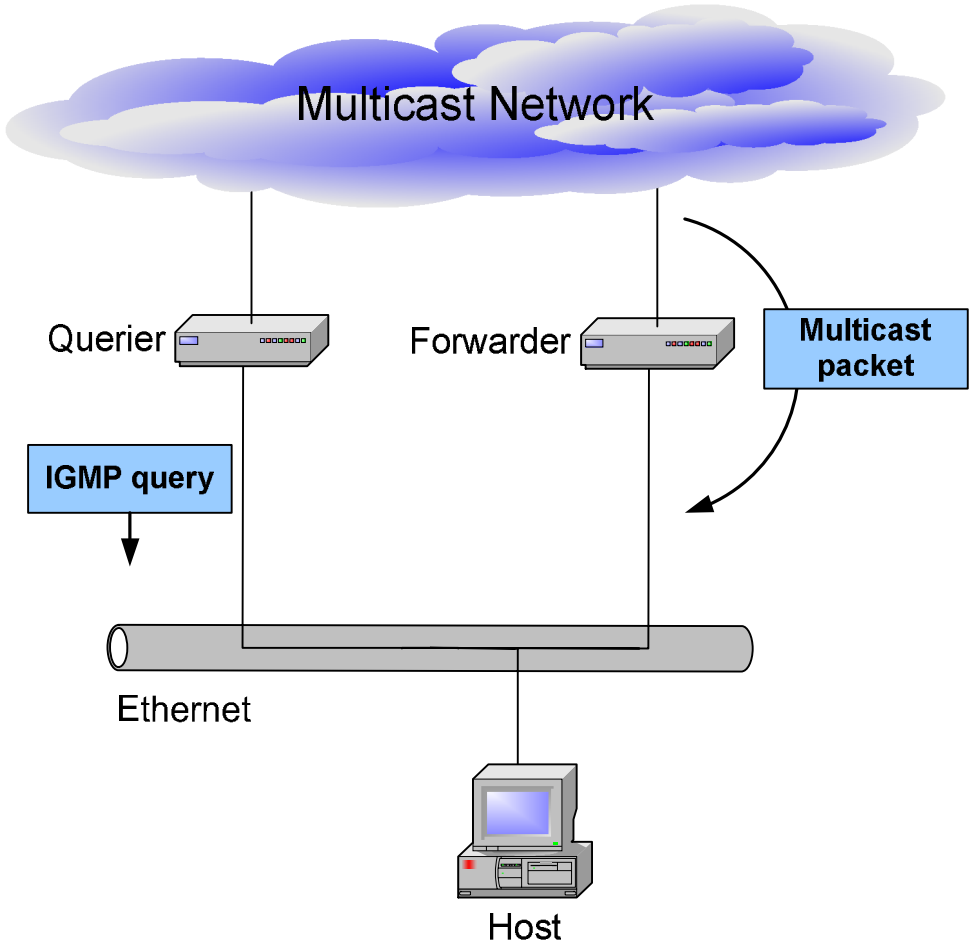
IGMP general query
IGMP group address = 0
Destination IP address = 224.0.0.1
Source IP address = router's IP address

IGMP group-specific query
IGMP group address = group address
Destination IP address = group address
Source IP address = router's IP address

IGMP membership report
IGMP group address = group address
Destination IP address= group address
Source IP address = host's IP address

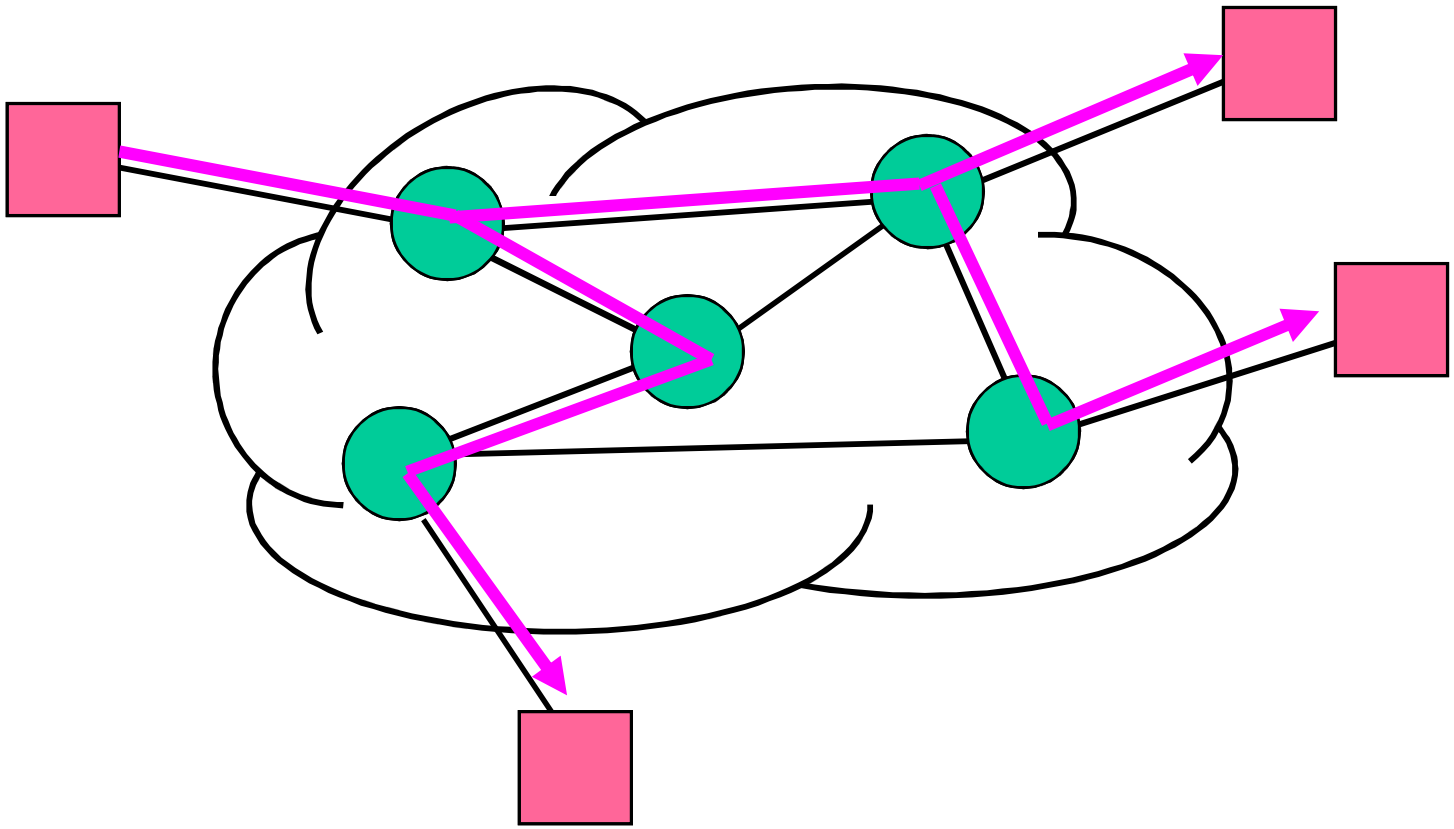
Networks with multiple multicast routers

- Only one router responds to IGMP queries (***Querier***)
  - Router with smallest IP address becomes the querier on a network.
- One router forwards multicast packets to the network (***Forwarder)***.



Multicast Routing Protocols

- **Goal:** Build a spanning tree between all members of a multicast group



DHCP

Host internet configuration

- We can always set manually for our machine:
  - **IP address**: choose whichever, but:
    - in case you choose one NOT on the local network, nothing can be routed to you
    - in case you choose existing IP on network, problems happen
  - **Netmask**: which bits in address are network, which host (often 255.255.255.0, but may vary)
  - **Default gateway**: IP address of the local router
  - **DNS server**: IP address of the server who to ask for IP addresses of server names

However ...

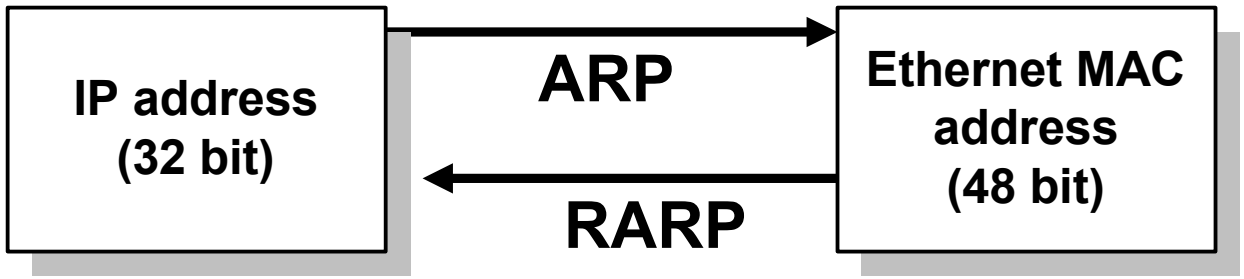
- Setting all this manually:
  - Requires knowing all this information
  - Is tedious anyway
- DHCP is the service normally (often) set up on the local network to make these settings automatically. We should have:
  - A server on the network must run DHCP service
  - Your machine must run a DHCP client daemon

Dynamic Assignment of IP addresses

- Dynamic assignment of IP addresses is desirable for several reasons:
  - IP addresses are assigned on-demand
  - Avoid manual IP configuration
  - Support mobility of laptops

Solutions for dynamic assignment of IP addresses

- **Reverse Address Resolution Protocol (RARP)**
  - Works similar to ARP
  - Broadcast a request for the IP address associated with a given MAC address
  - RARP server responds with an IP address
  - Only assigns IP address (not the default router and subnetmask)



BOOTP

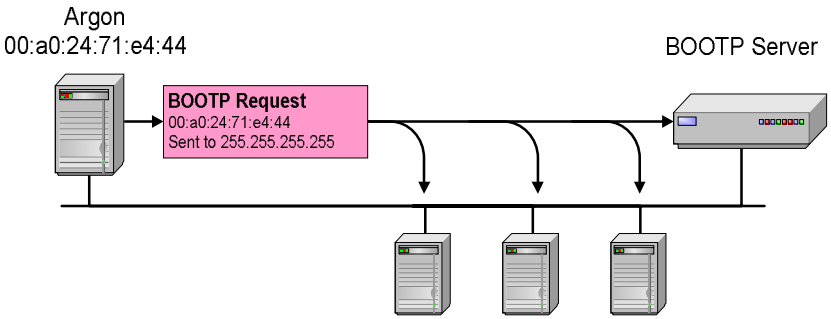
- **BOOTstrap Protocol (BOOTP)**
  - From 1985
  - Host can configure its IP parameters at boot time.
  - 3 services.
    - IP address assignment.
    - Detection of the IP address for a serving machine.
    - The name of a file to be loaded and executed by the client machine (boot file name)

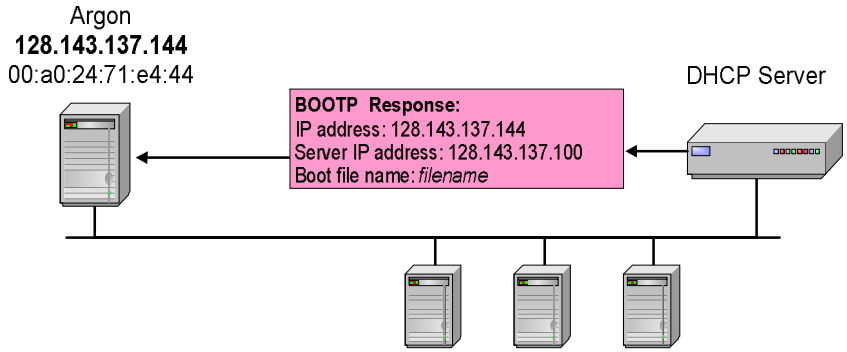
  - Not only assign IP address, but also default router, network mask, etc.
  - Sent as UDP messages (UDP Port 67 (server) and 68 (host))
  - Use limited broadcast address (255.255.255.255):
    - These addresses are never forwarded

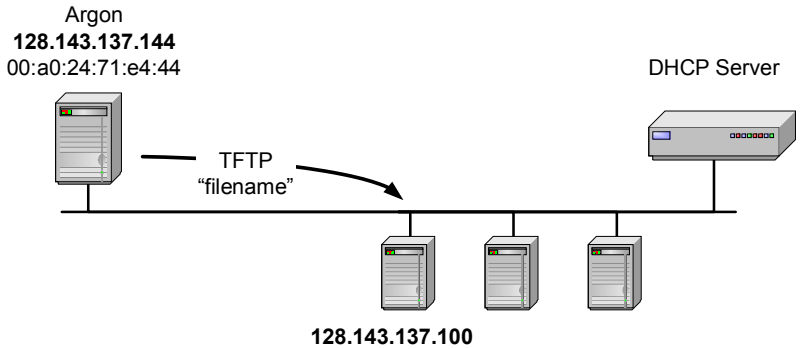
DHCP

- **Dynamic Host Configuration Protocol (DHCP)**
  - From 1993
  - An extension of BOOTP
  - Same port numbers as BOOTP
  - Extensions:
    - Supports temporary allocation (“leases”) of IP addresses
    - DHCP client can acquire all IP configuration parameters needed to operate
  - DHCP is the preferred mechanism for dynamic assignment of IP addresses
  - DHCP can interoperate with BOOTP clients.

BOOTP Interaction

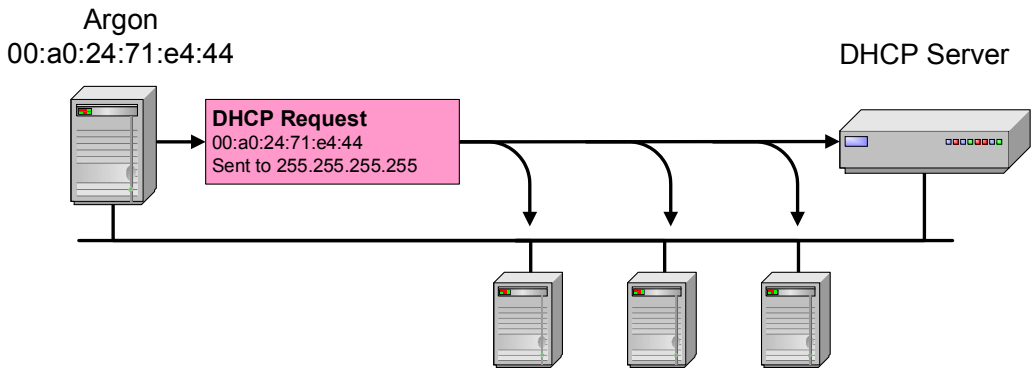


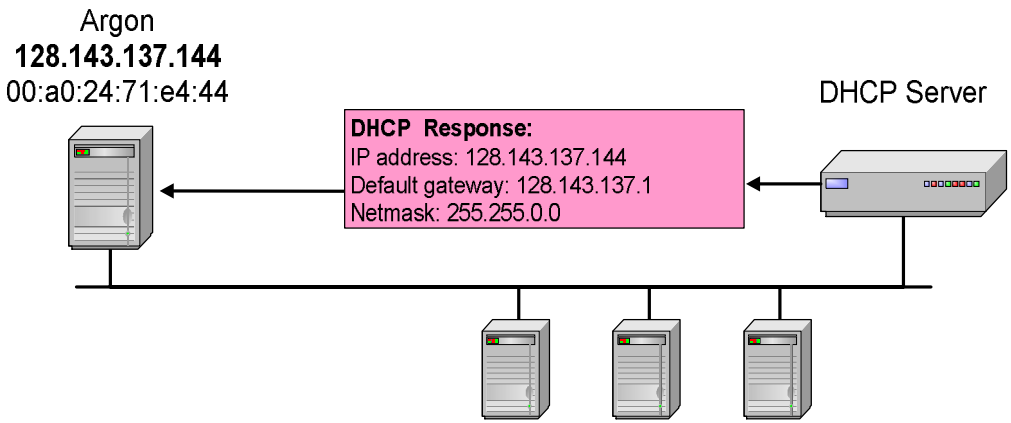




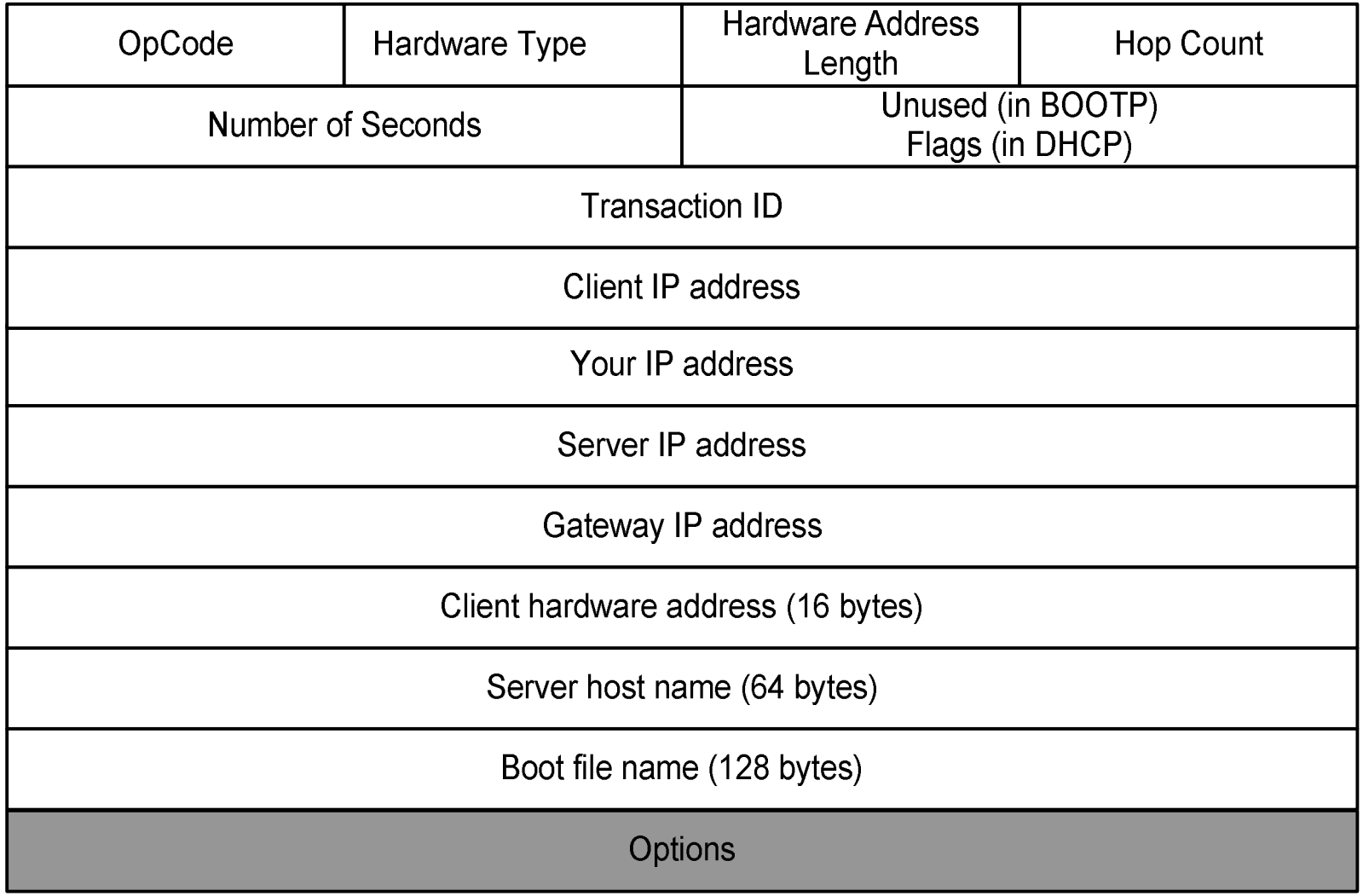
- BOOTP can be used for downloading memory image for diskless workstations
- Assignment of IP addresses to hosts is static

DHCP Interaction (simplified)





BOOTP/DHCP Message Format



BOOTP/DHCP

- **OpCode**: *1 (Request), 2(Reply)*
  Note: DHCP message type is sent in an option
- **Hardware Type**: *1 (for Ethernet)*
- **Hardware address length**: *6 (for Ethernet)*
- **Hop count**: *set to 0 by client*
- **Transaction ID**: *Integer (used to match reply to response)*
- **Seconds:** *number of seconds since the client started to boot*
- **Client IP address, Your IP address, server IP address, Gateway IP address, client hardware address, server host name, boot file name:**
  client fills in the information that it has, leaves rest blank

DHCP Message Type

- Message type is sent as an option.

Value	Message Type
1	DHCPDISCOVER
2	DHCPOFFER
3	DHCPREQUEST
4	DHCPDECLINE
5	DHCPACK
6	DHCPNAK
7	DHCPRELEASE
8	DHCPINFORM

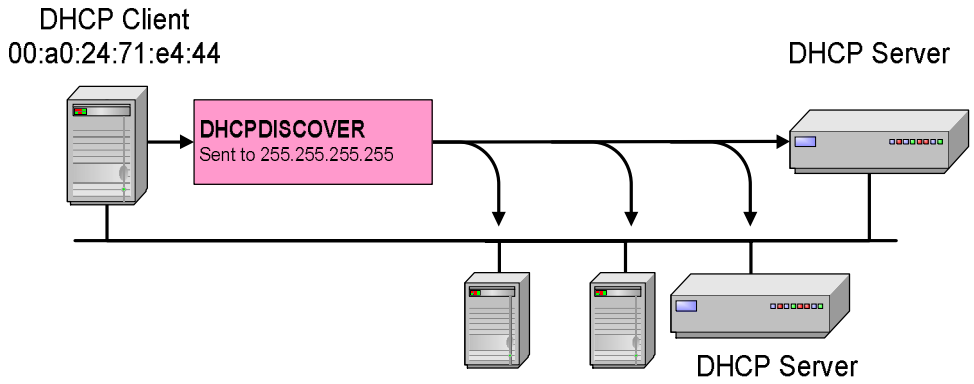
Other options (selection)

- Other DHCP information that is sent as an option:

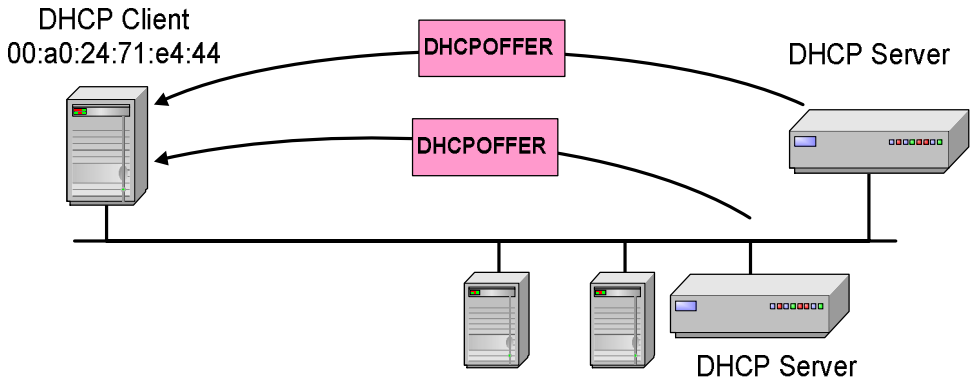
  Subnet Mask, Name Server, Hostname, Domain Name, Forward On/Off, Default IP TTL, Broadcast Address, Static Route, Ethernet Encapsulation, X Window Manager, X Window Font, DHCP Msg Type, DHCP Renewal Time, DHCP Rebinding, Time SMTP-Server, SMTP-Server, Client FQDN, Printer Name, …

DHCP Operation

- DCHP DISCOVER

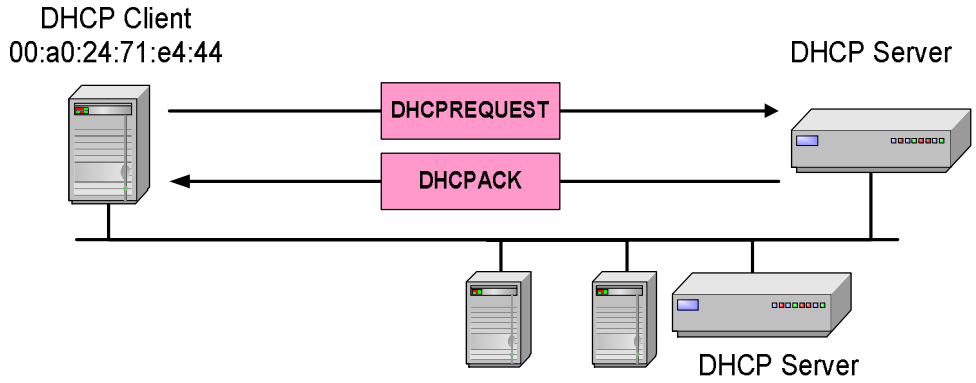


DCHP OFFER

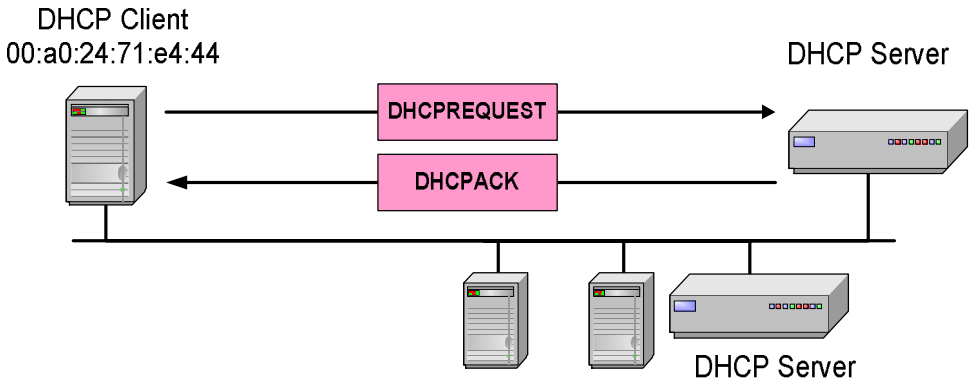


DHCP Operation

DCHP DISCOVER



At this time, the DHCP client can start to use the IP address



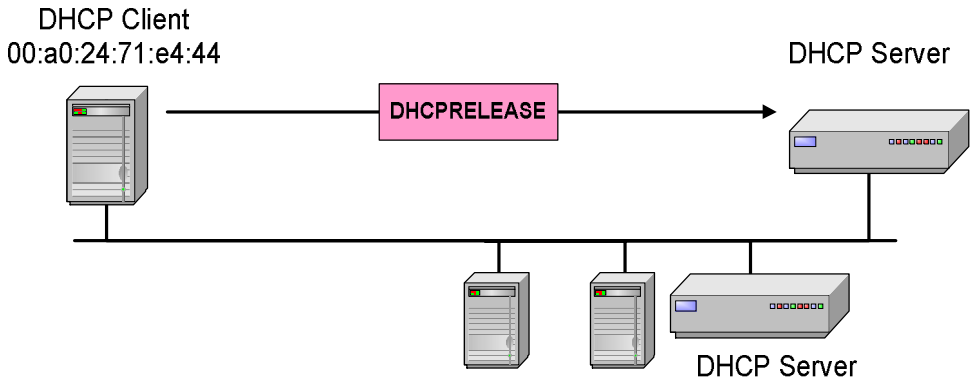
Renewing a Lease

(sent when 50% of lease has expired)

If DHCP server sends DHCPNACK, then address is released.

DHCP Operation

DCHP RELEASE



At this time, the DHCP client has released the IP address

DNS

Domain names and IP addresses

- People prefer to use easy-to-remember names instead of IP addresses

- Domain names are alphanumeric names for IP addresses e.g., neon.ece.utoronto.ca, www.google.com, ietf.org
- The domain name system (DNS) is an Internet-wide distributed database that translates between domain names and IP addresses

- **How important is DNS?**

  Imagine what happens when the local DNS server is down.

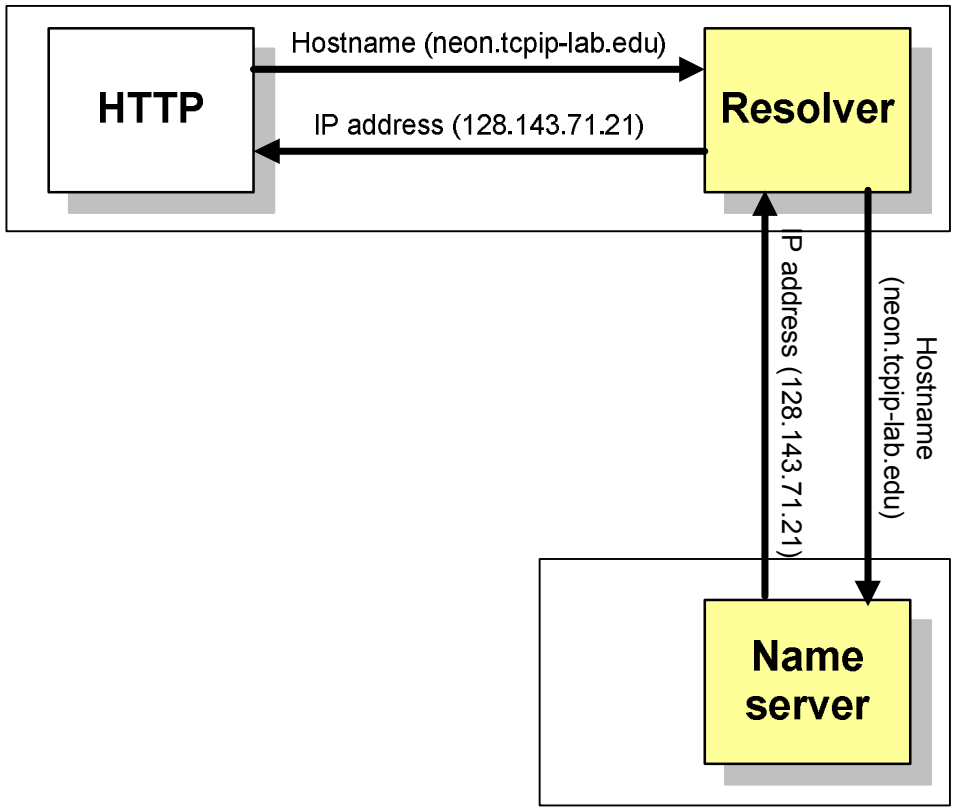
Before there was DNS ….

…. there was the HOSTS.TXT file

- Before DNS (until 1985), the name-to-IP address was done by downloading a single file (hosts.txt) from a central server with FTP.
  - Names in hosts.txt are not structured.
  - The hosts.txt file still works on most operating systems. It can be used to define local names.
  - Linux: /etc/hosts

Resolver and name server

1. An application program on a host accesses the domain system through a DNS client, called the **resolver**
2. Resolver contacts DNS server, called name server
3. DNS server returns IP address to resolver which passes the IP address to application



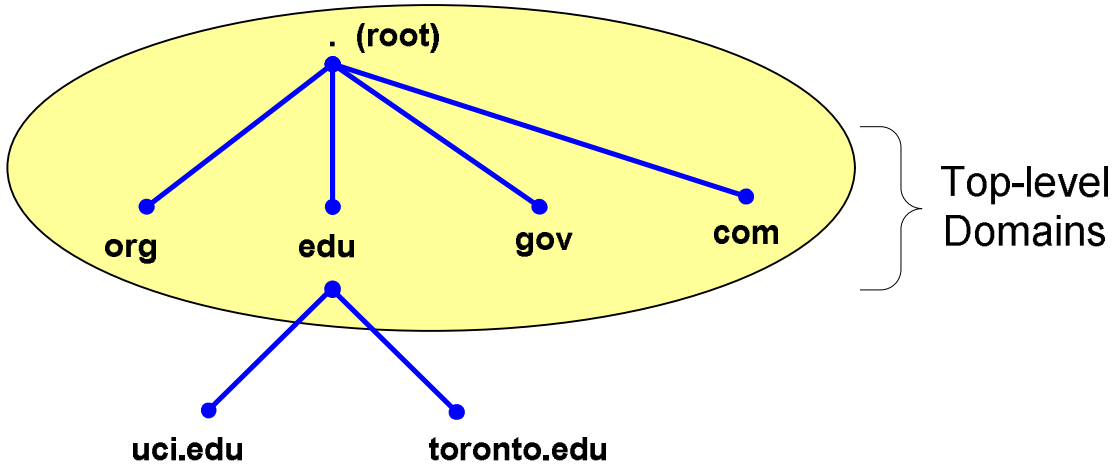
- Reverse lookups are also possible, i.e., find the hostname given an IP address

Design principle of DNS

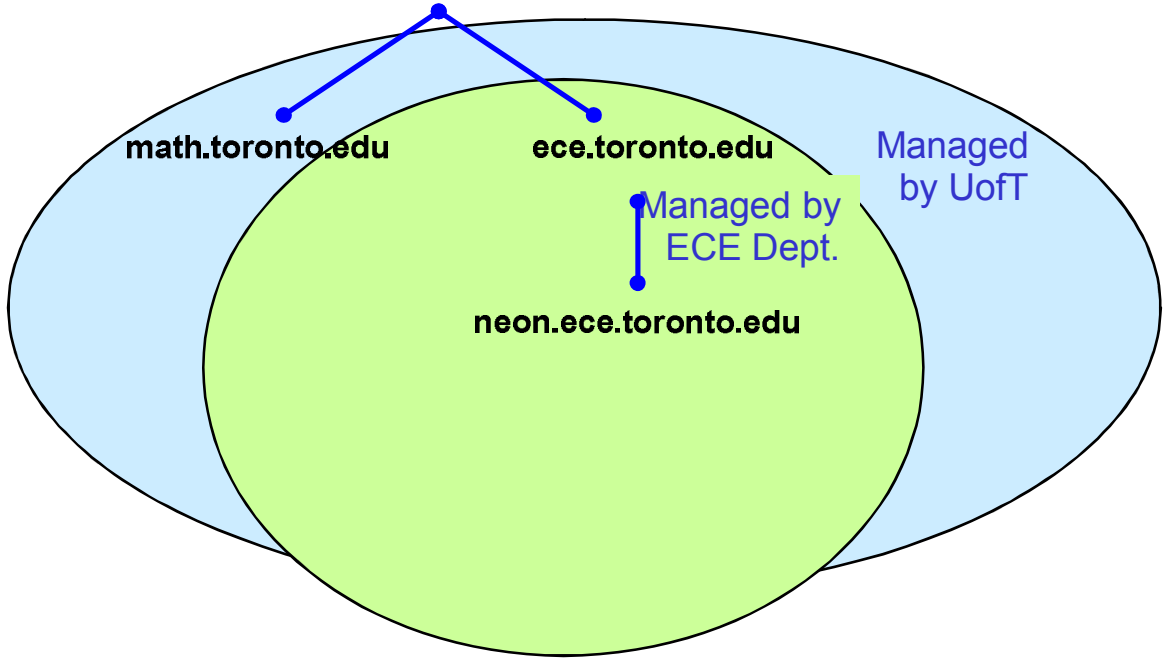
- The naming system on which DNS is based is a hierarchical and logical tree structure called the *domain namespace*.
- An organization obtains authority for parts of the name space, and can add additional layers of the hierarchy
- Names of hosts can be assigned without regard of location on a link layer network, IP network or autonomous system
- In practice, allocation of the domain names generally follows the allocation of IP address, e.g.,
  - All hosts with network prefix 128.143/16 have domain name suffix virginia.edu
  - All hosts on network 128.143.136/24 are in the Computer Science Department of the University of Virginia

DNS Name hierarchy

- DNS hierarchy can be represented by a tree
- Root and top-level domains are administered by an Internet central name registration authority (ICANN)



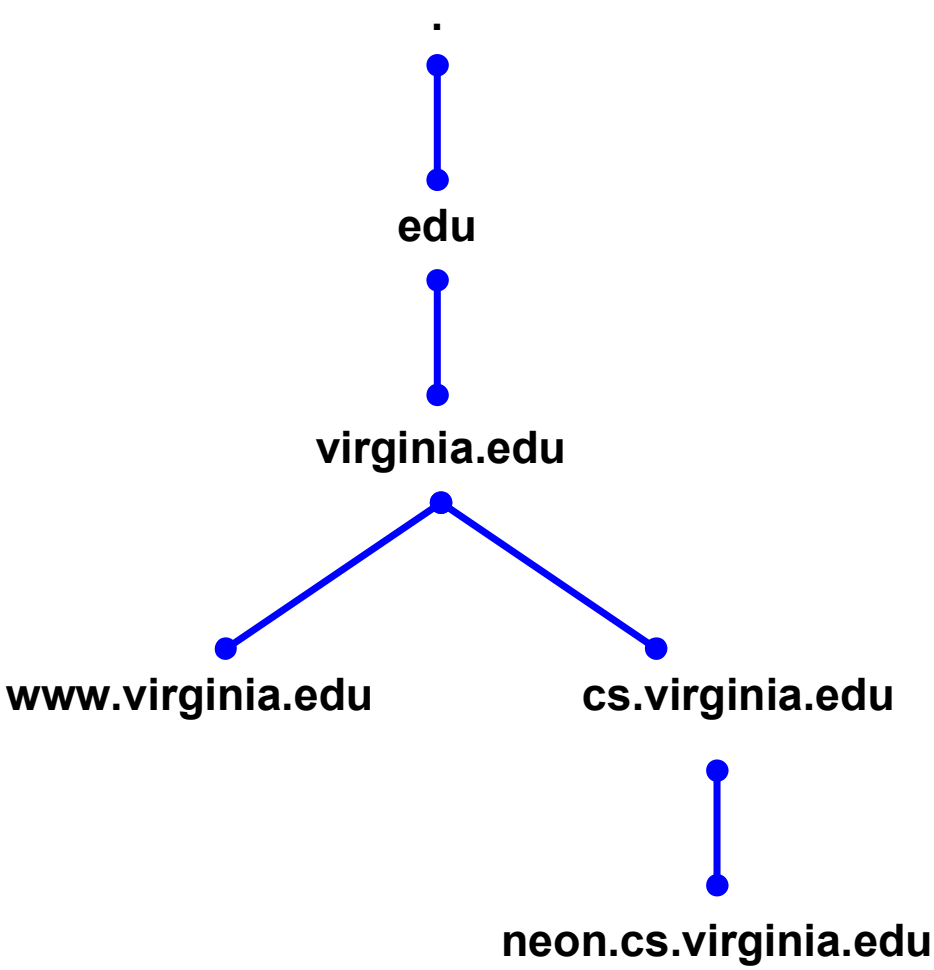
- Below top-level domain, administration of name space is delegated to organizations
- Each organization can delegate further



Domain name system

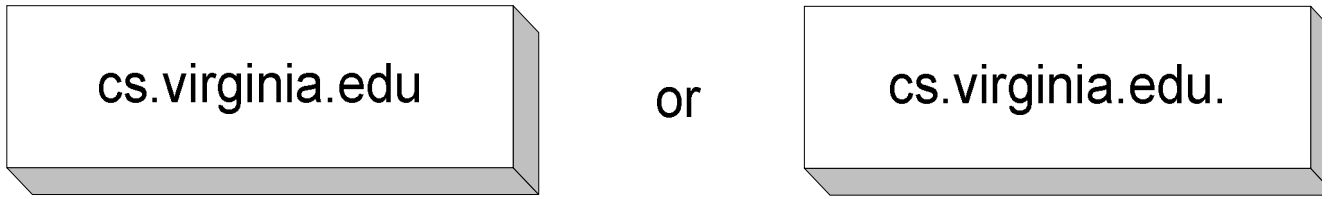
- Each node in the DNS tree represents a DNS name
- Each branch below a node is a DNS domain.
  - DNS domain can contain hosts or other domains (subdomains)

- Example:
  DNS domains are
  ., edu, virginia.edu, cs.virginia.edu



Domain names

- Hosts and DNS domains are named based on their position in the domain tree
- Every node in the DNS domain tree can be identified by a unique Fully Qualified Domain Name (FQDN). The FQDN gives the position in the DNS tree.



- A FQDN consists of labels (“cs”,“virginia”,”edu”) separated by a period (“.”)
- There can be a period (“.”) at the end.
- Each label can be up to 63 characters long
- FQDN contains characters, numerals, and dash character (“-”)
- FQDNs are not case-sensitive

Top-level domains

- Three types of top-level domains:
  - Organizational: 3-character code indicates the function of the organization
    - Used primarily within the US
    - Examples: gov, mil, edu, org, com, net
  - Geographical: 2-character country or region code
    - Examples: us, va, jp, de
  - Reverse domains: A special domain (in-addr.arpa) used for IP address-to-name mapping

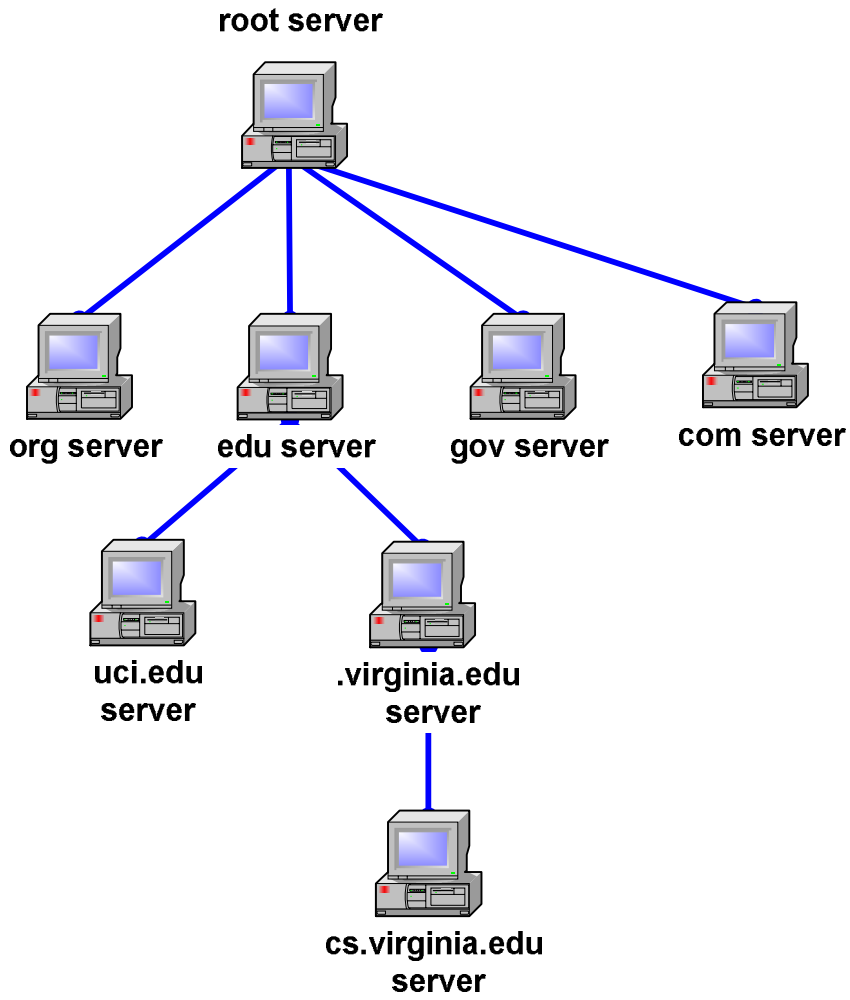
There are more than 200 top-level domains.

Organizational top-level domains

com	Commercial organizations
edu	Educational institutions
gov	Government institutions
int	International organizations
mil	U.S. military institutions
net	Networking organizations
org	Non-profit organizations

Hierarchy of name servers

- The resolution of the hierarchical name space is done by a hierarchy of name servers
- Each server is responsible (authoritative) for a contiguous portion of the DNS namespace, called a zone.
- Zone is a part of the subtree
- DNS server answers queries about hosts in its zone



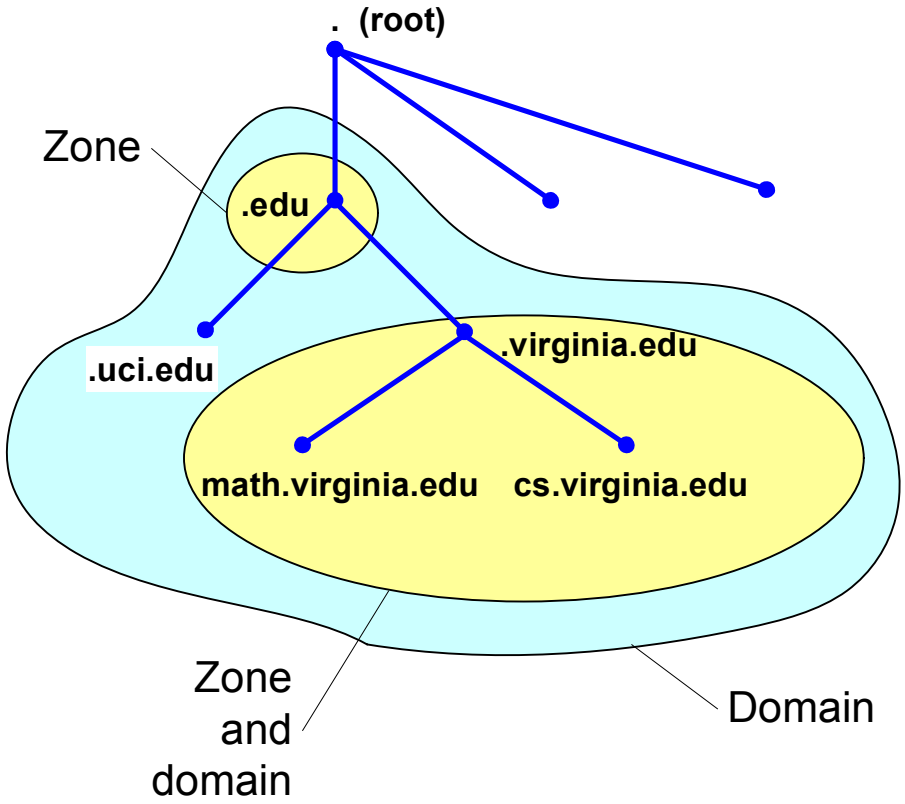
Authority and delegation

- Authority for the root domain is with the Internet Corporation for Assigned Numbers and Names (ICANN)
- ICANN delegates to accredited registrars (for gTLDs) and countries for country code top level domains (ccTLDs)
- Authority can be delegated further

- Chain of delegation can be obtained by reading domain name from right to left.
- Unit of delegation is a “zone”.

DNS domain and zones

- Each zone is anchored at a specific domain node, but zones are not domains.
- *A DNS domain* is a branch of the namespace
- A zone is a portion of the DNS namespace generally stored in a file (It could consists of multiple nodes)
- A server can divide part of its zone and delegate it to other servers



Primary and secondary name servers

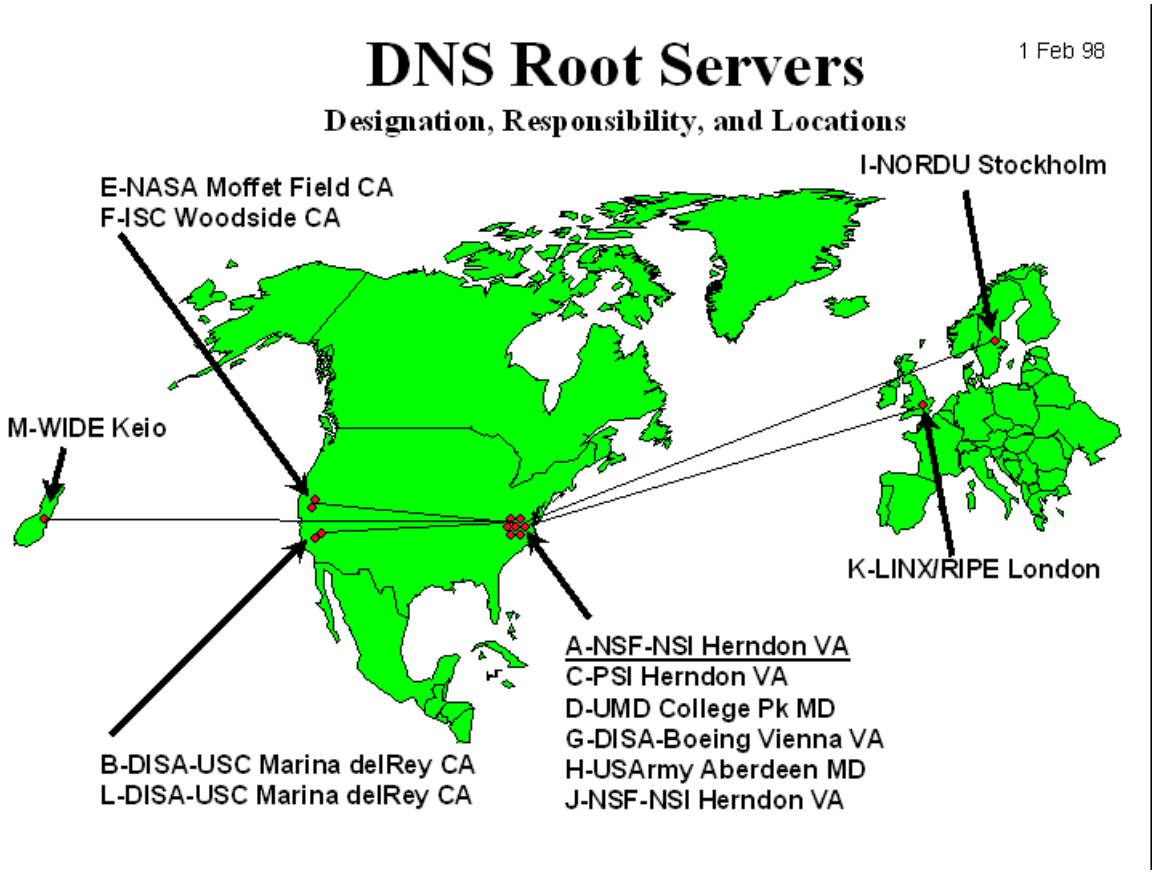
- For each zone, there must be a primary name server and a secondary name server
  - The primary server (master server) maintains a zone file which has information about the zone. Updates are made to the primary server
  - The secondary server copies data stored at the primary server.

Adding a host:

- When a new host is added (“gold.cs.virginia.edu”) to a zone, the administrator adds the IP information on the host (IP address and name) to a configuration file on the primary server

Root name servers

- The root name servers know how to find the authoritative name servers for all top-level zones.
- There are only 13 root name servers
- Root servers are critical for the proper functioning of name resolution

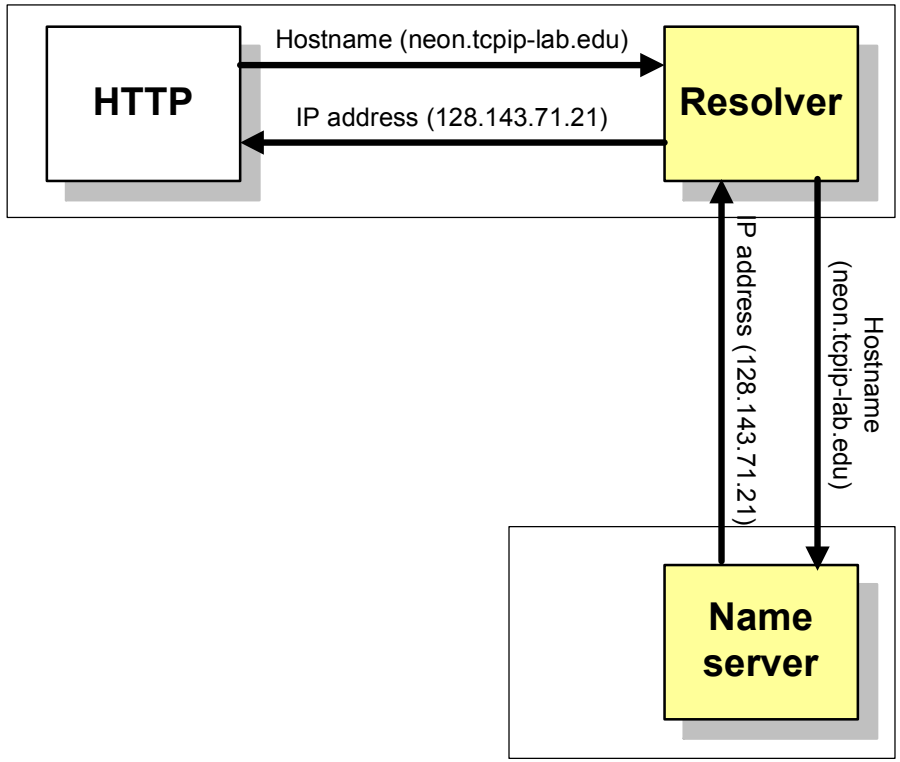


Addresses of root servers

A.ROOT-SERVERS.EDU.	(formerly NS.INTERNIC.NET)	10.0.2.32
A.ROOT-SERVERS.NET.	(formerly NS1.ISI.EDU)	198.41.0.4
B.ROOT-SERVERS.NET.	(formerly C.PSI.NET)	128.9.0.107
C.ROOT-SERVERS.NET.	(TERP.UMD.EDU)	192.33.4.12
D.ROOT-SERVERS.NET.	(NS.NASA.GOV)	128.8.10.90
E.ROOT-SERVERS.NET.	(NS.ISC.ORG)	192.203.23
F.ROOT-SERVERS.NET.	(NS.NIC.DDN.MIL)	192.5.5.241
G.ROOT-SERVERS.NET.	(AOS.ARL.ARMY.MIL)	192.112.36.4
H.ROOT-SERVERS.NET.	(NIC.NORDU.NET)	128.63.2.53
I.ROOT-SERVERS.NET.	(at NSI (InterNIC))	192.36.148.17
J.ROOT-SERVERS.NET.	(operated by RIPE NCC)	198.41.0.10
K.ROOT-SERVERS.NET.	(at ISI (IANA))	193.0.14.129
L.ROOT-SERVERS.NET.	(operated by WIDE, Japan)	198.32.64
M.ROOT-SERVERS.NET.		202.12.27.33

Domain name resolution

1. User program issues a request for the IP address of a hostname
2. Local resolver formulates a DNS query to the name server of the host
3. Name server checks if it is authorized to answer the query.
   a) If yes, it responds.
   b) Otherwise, it will query other name servers, starting at the root tree
4. When the name server has the answer it sends it to the resolver.



Recursive and Iterative Queries

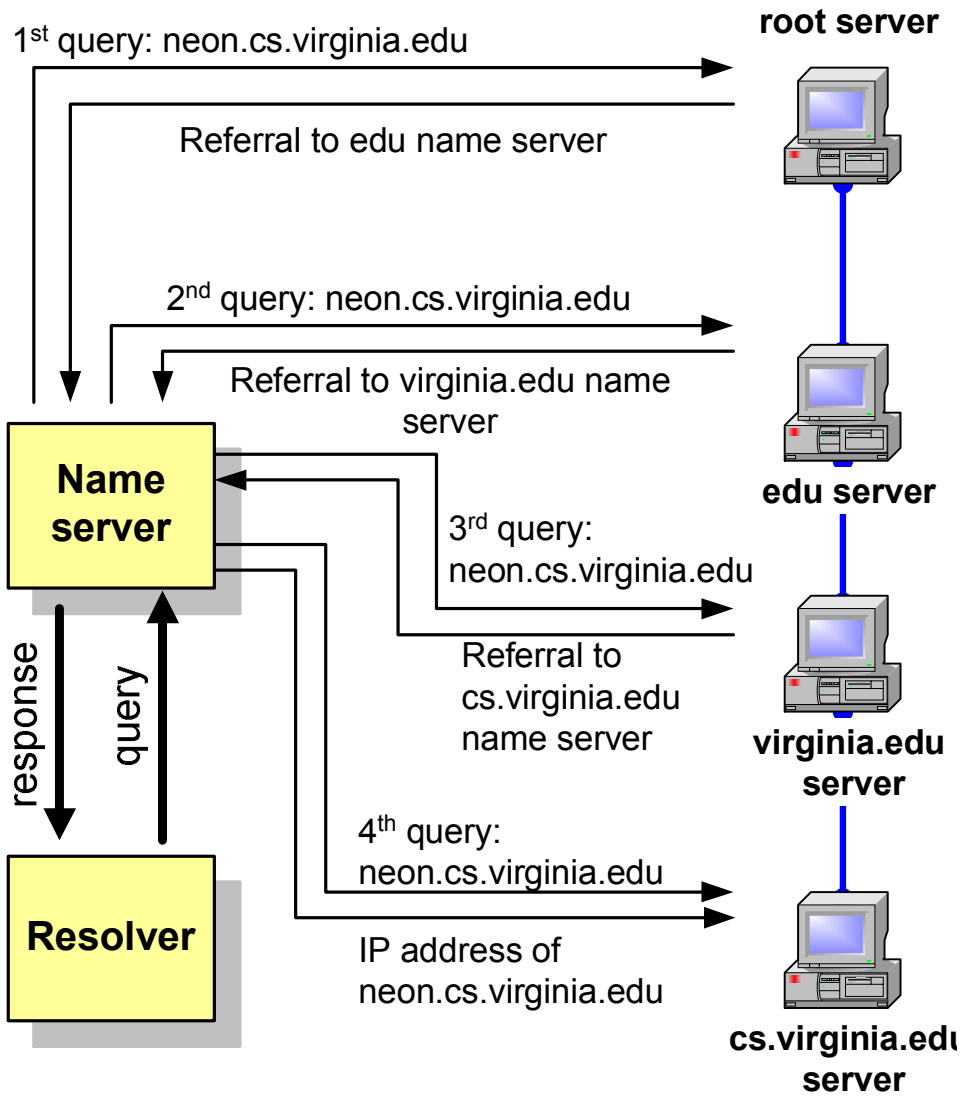
- There are two types of queries:
  - Recursive queries
  - Iterative (non-recursive) queries

- The type of query is determined by a bit in the DNS query

- Recursive query: When the name server of a host cannot resolve a query, the server issues a query to resolve the query
- Iterative queries: When the name server of a host cannot resolve a query, it sends a referral to another server to the resolver

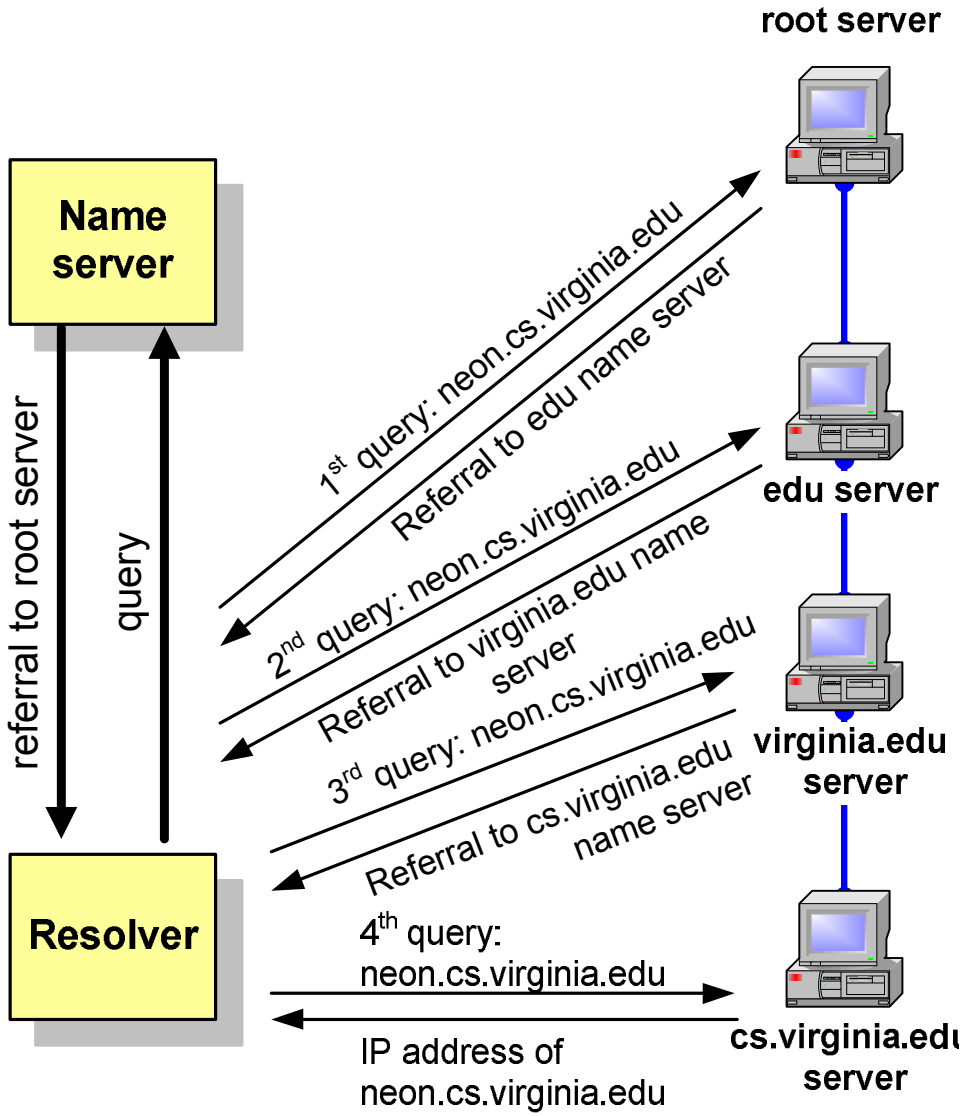
Recursive queries

- In a recursive query, the resolver expects the response from the name server
- If the server cannot supply the answer, it will send the query to the “closest known” authoritative name server (here: In the worst case, the closest known server is the root server)
- The root sever sends a referral to the “edu” server. Querying this server yields a referral to the server of “virginia.edu”
- … and so on



Iterative queries

- In an iterative query, the name server sends a closest known authoritative name server a referral to the root server.
- This involves more work for the resolver



Caching

- To reduce DNS traffic, name servers caches information on domain name/IP address mappings
- When an entry for a query is in the cache, the server does not contact other servers
- Note: If an entry is sent from a cache, the reply from the server is marked as “unauthoritative”

Resource Records

- The database records of the distributed data base are called resource records (RR)
- Resource records are stored in configuration files (zone files) at name servers.
- Left Resource records for a zone:

db.mylab.com

```
$TTL 86400
mylab.com.  IN SOA PC4.mylab.com.
                   hostmaster.mylab.com. (
                   1 ; serial
                   28800 ; refresh
                   7200 ; retry
                   604800 ; expire
                   86400 ; ttl
                   )

;
mylab.com.   IN    NS    PC4.mylab.com.
;
localhost          A     127.0.0.1
PC4.mylab.com.     A     10.0.1.41
PC3.mylab.com.     A     10.0.1.31
PC2.mylab.com.     A     10.0.1.21
PC1.mylab.com.     A     10.0.1.11
```

Resource Records

db.mylab.com

```
$TTL 86400
mylab.com. IN SOA PC4.mylab.com. hostmaster.mylab.com. (
                  1 ; serial
                  28800 ; refresh
                  7200 ; retry
                  604800 ; expire
                  86400 ; ttl
                  )

;
mylab.com.  IN    NS    PC4.mylab.com.
;
localhost         A     127.0.0.1
PC4.mylab.com.    A     10.0.1.41
PC3.mylab.com.    A     10.0.1.31
PC2.mylab.com.    A     10.0.1.21
PC1.mylab.com.    A     10.0.1.11
```

Max. age of cached data in seconds

Start of authority (SOA) record.
Means: "This name server is authoritative for the zone Mylab.com"
* PC4.mylab.com is the name server
* hostmaster@mylab.com is the email address of the person in charge

Name server (NS) record.
One entry for each authoritative name server

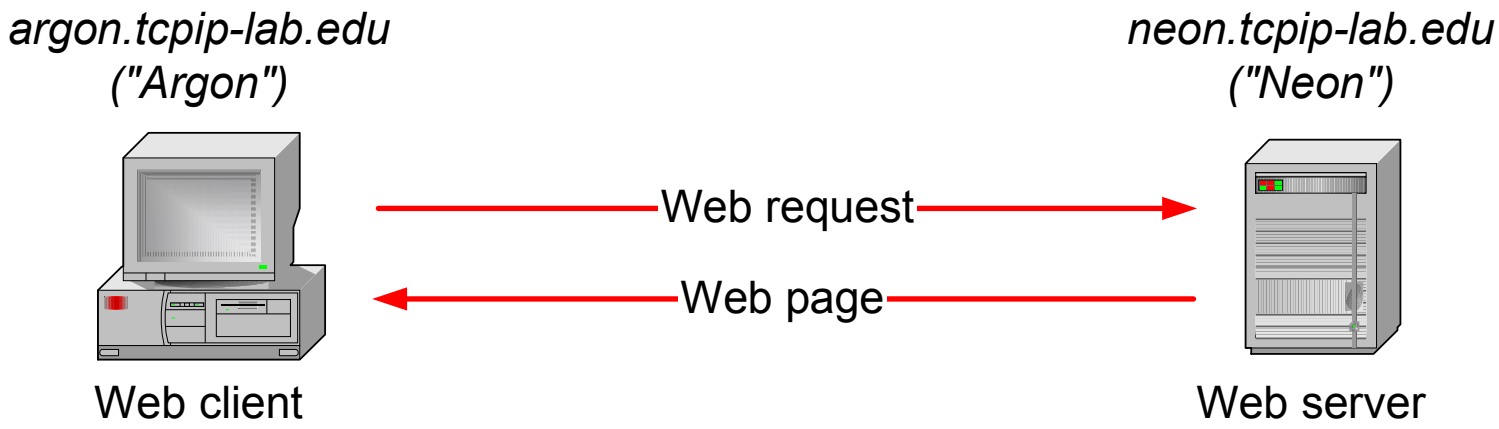
Address (A) records.
One entry for each hostaddress

Summary example

A simple TCP/IP Example

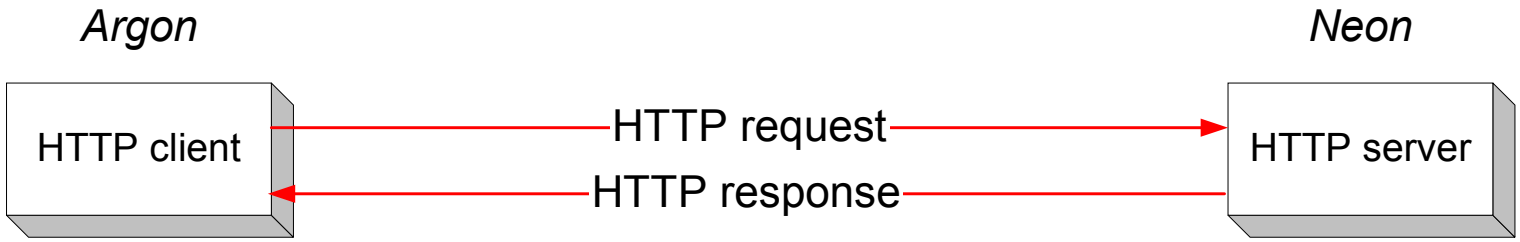
- A user on host *argon.tcpip-lab.edu* (“*Argon*”) makes a web access to URL

  http://neon.tcpip-lab.edu/index.html.



HTTP Request and HTTP response

- Web browser runs an HTTP client program
- Web server runs an HTTP server program
- HTTP client sends an HTTP request to HTTP server
- HTTP server responds with HTTP response



HTTP Request

```
GET /index.html HTTP/1.1
Accept: image/gif, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Host: neon.tcpip-lab.edu
Connection: Keep-Alive
```

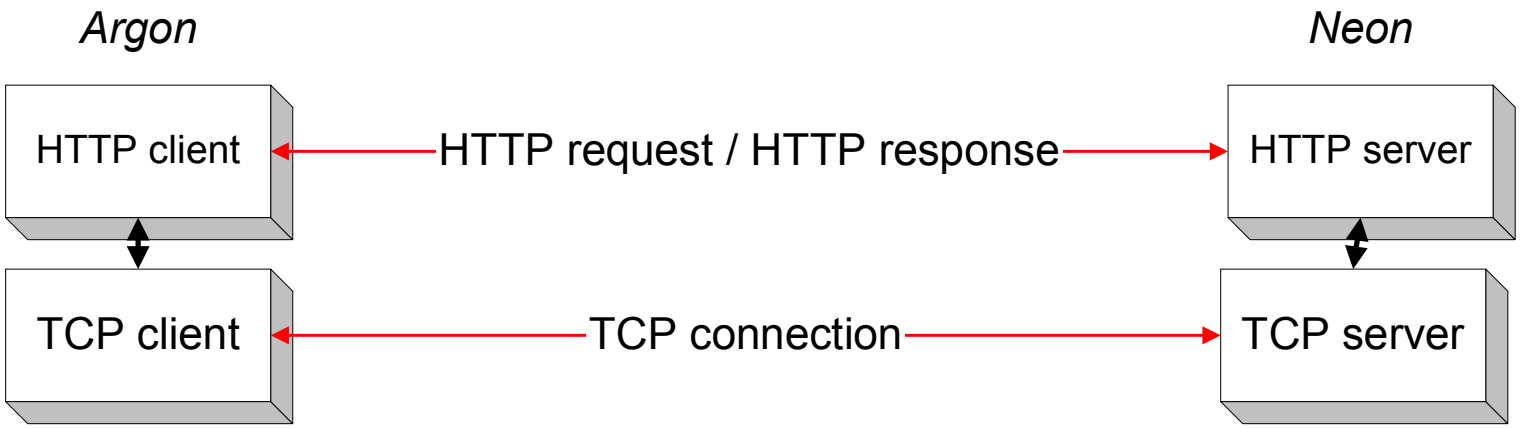
HTTP Response

```
HTTP/1.1 200 OK
Date: Sat, 25 May 2002 21:10:32 GMT
Server: Apache/1.3.19 (Unix)
Last-Modified: Sat, 25 May 2002 20:51:33 GMT
ETag: "56497-51-3ceff955"
Accept-Ranges: bytes
Content-Length: 81
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

<HTML>
<BODY>
<H1>Internet Lab</H1>
Click <a href="http://www.tcpip-
lab.net/index.html">here</a> for the Internet Lab webpage.
</BODY>
</HTML>
```

From HTTP to TCP

- To send a request, the HTTP client program establishes an TCP connection to the HTTP server at Neon.
- The HTTP server at Neon has a TCP server running



Resolving hostnames and port numbers

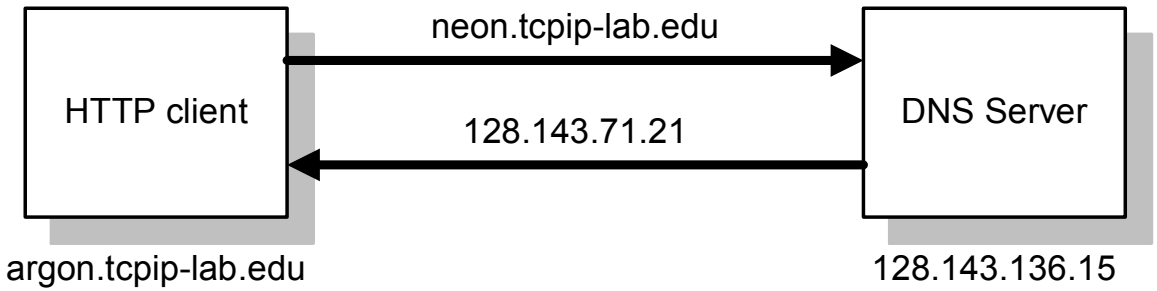
- Since TCP does not work with hostnames and also does not know how to find the HTTP server program at Neon, two things must happen:

1. The name “neon.tcpip-lab.edu” must be translated into a 32-bit **IP address.**

2. The HTTP server at Neon must be identified by a 16-bit **port number**.

Translating a hostname into an IP address

- The translation of the hostname *neon.tcpip-lab.edu* into an IP address is done via a database lookup



- The distributed database used is called the ***Domain Name System (DNS)***
- All machines on the Internet have an IP address:

argon.tcpip-lab.edu	*128.143.137.144*
neon.tcpip-lab.edu	*128.143.71.21*

Finding the port number

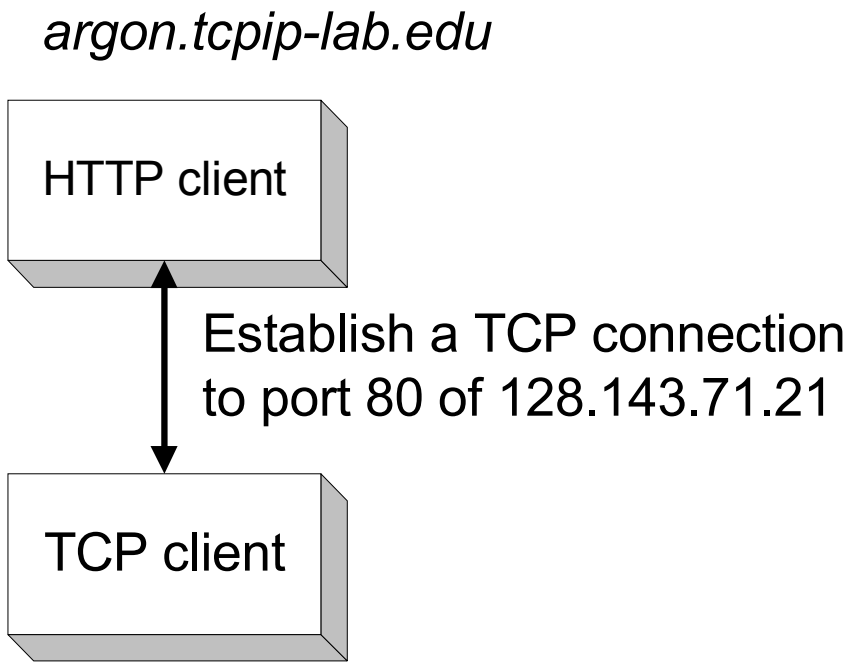
- **Note:** Most services on the Internet are reachable via well-known ports. E.g. All HTTP servers on the Internet can be reached at port number "80".
- **So:** Argon simply knows the port number of the HTTP server at a remote machine.

- On most Unix systems, the well-known ports are listed in a file with name /etc/services. The well-known port numbers of some of the most popular services are:

ftp	21	finger	79
telnet	23	http	80
smtp	25	nntp	119

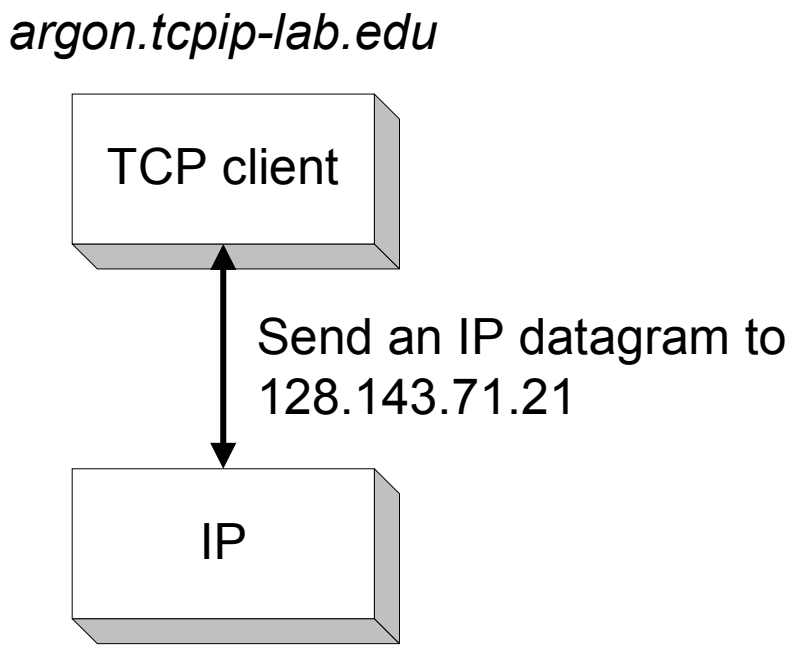
Requesting a TCP Connection

- The HTTP client at *argon.tcpip-lab.edu* requests the TCP client to establish a connection to port 80 of the machine with address 128.141.71.21



Invoking the IP Protocol

- The TCP client at *Argon* sends a request to establish a connection to port 80 at *Neon*
- This is done by asking its local IP module to send an IP datagram to *128.143.71.21*
- *(The data portion of the IP datagram contains the request to open a connection)*



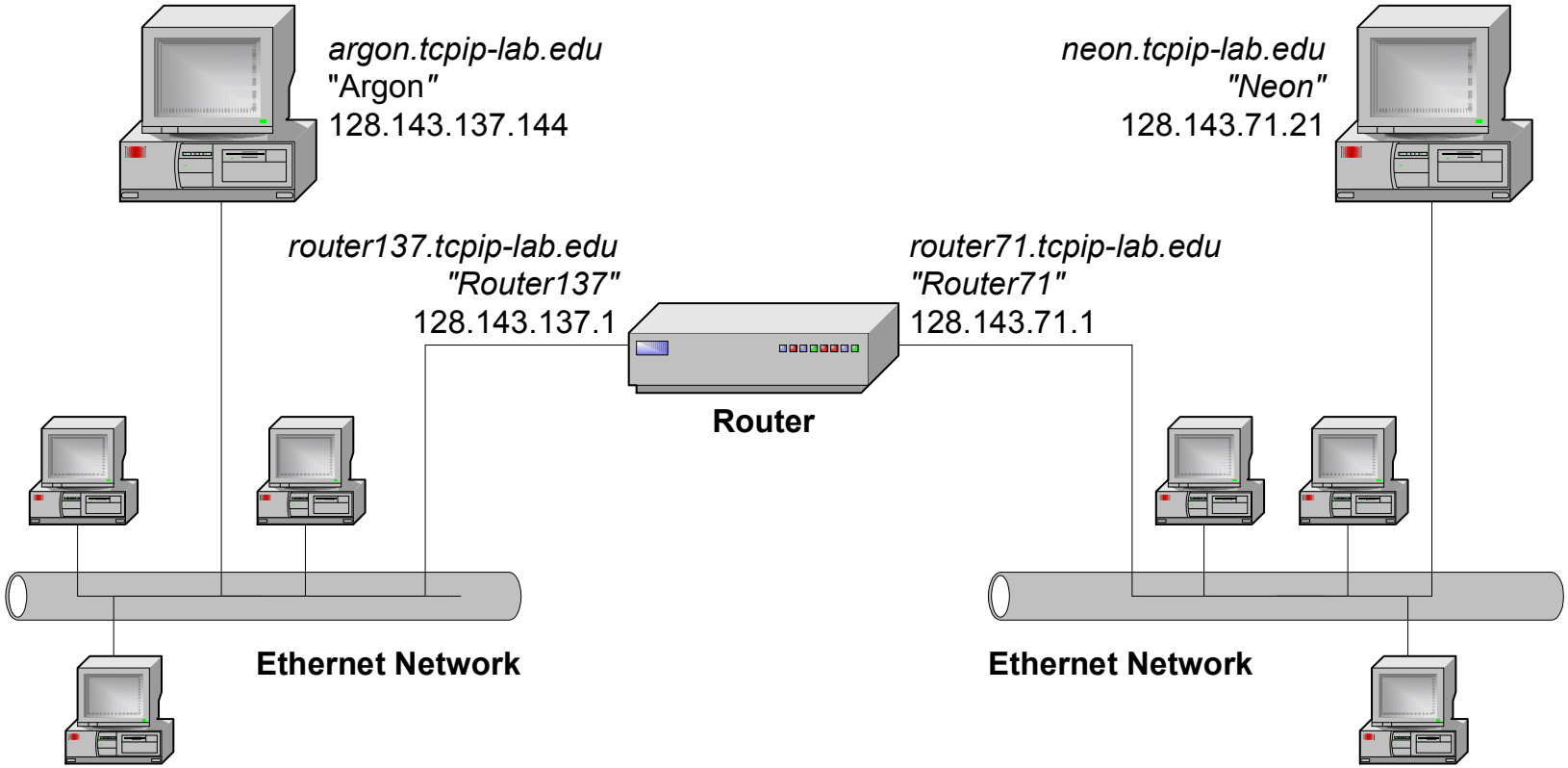
Sending the IP datagram to an IP router

- *Argon (128.143.137.144)* can deliver the IP datagram directly to *Neon (128.143.71.21*), only if it is on the same IP network (sometimes called "subnet").

- But *Argon* and N*eon* are not on the same IP network
  (**Q**: How does *Argon* know this?)

- So, *Argon* sends the IP datagram to its default gateway
- The default gateway is an IP router
- The default gateway for *Argon* is *Router137.tcpip-lab.edu* (128.143.137.1).

The route from *Argon* to *Neon*

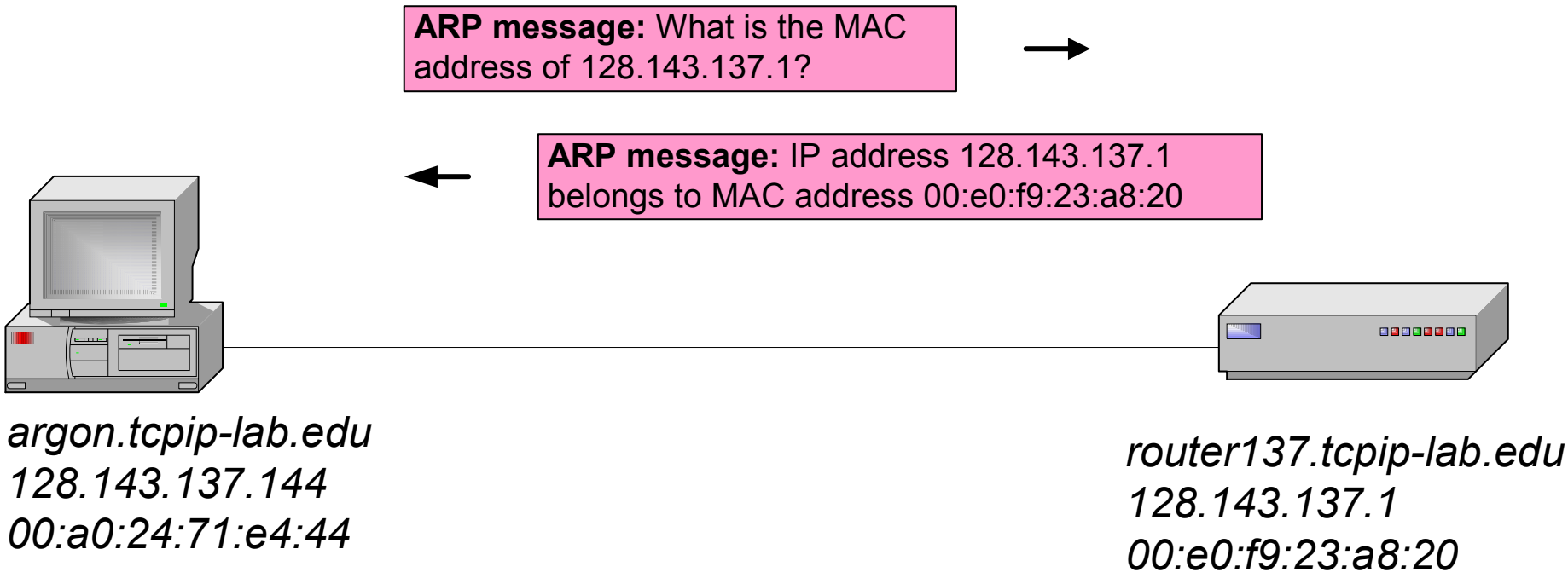


- Note that the gateway has a different name for each of its interfaces.

Finding the MAC address of the gateway

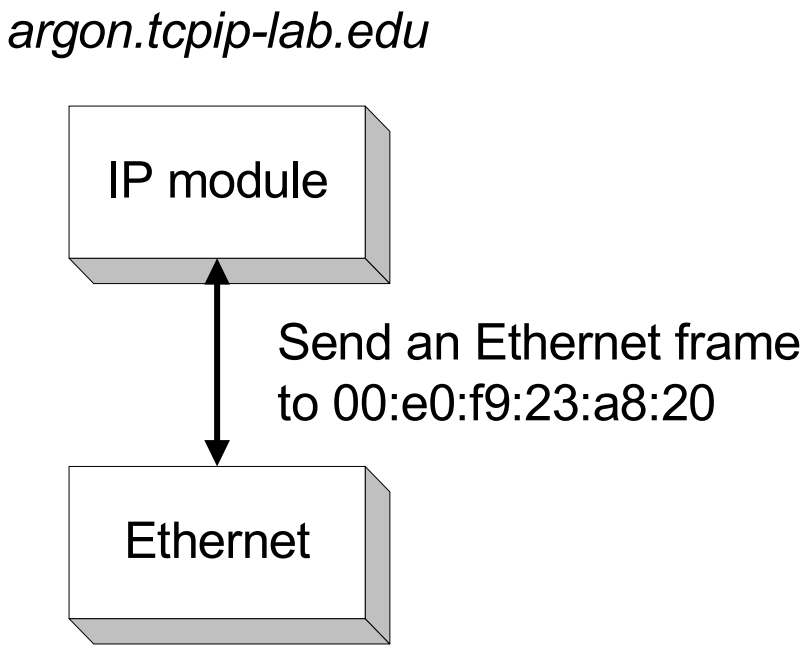
- To send an IP datagram to Router137, *Argon* puts the IP datagram in an Ethernet frame, and transmits the frame.
- However, Ethernet uses different addresses, so-called Media Access Control (MAC) addresses (also called: physical address, hardware address)
- Therefore, *Argon* must first translate the IP address 128.143.137.1 into a MAC address.
- The translation of addressed is performed via the Address Resolution Protocol (ARP)

Address resolution with ARP



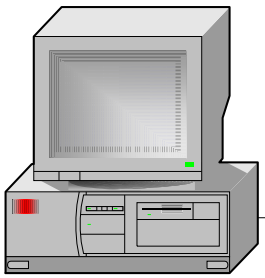
Invoking the device driver

- The IP module at *Argon*, tells its Ethernet device driver to send an Ethernet frame to address *00:e0:f9:23:a8:20*



Sending an Ethernet frame

- The Ethernet device driver of *Argon* sends the Ethernet frame to the Ethernet network interface card (NIC)
- The NIC sends the frame onto the wire



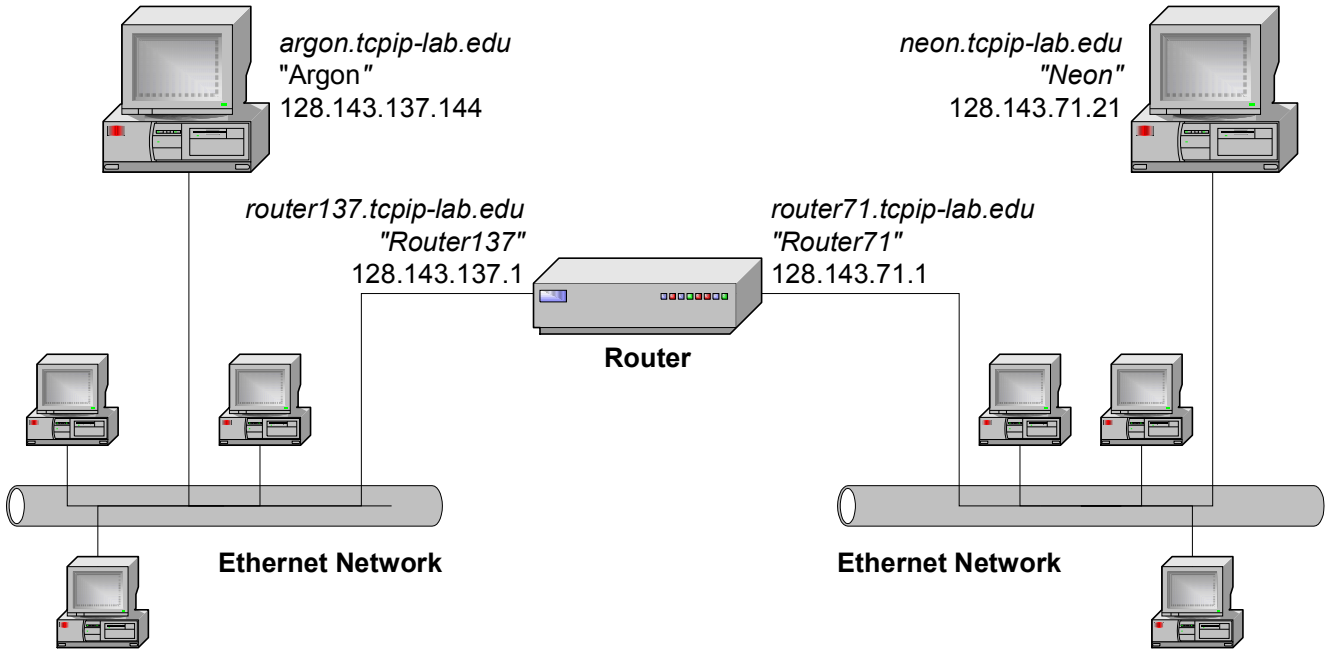
IP Datagram for Neon

argon.tcpip-lab.edu
128.143.137.144
00:a0:24:71:e4:44

router137.tcpip-lab.edu
128.143.137.1
00:e0:f9:23:a8:20

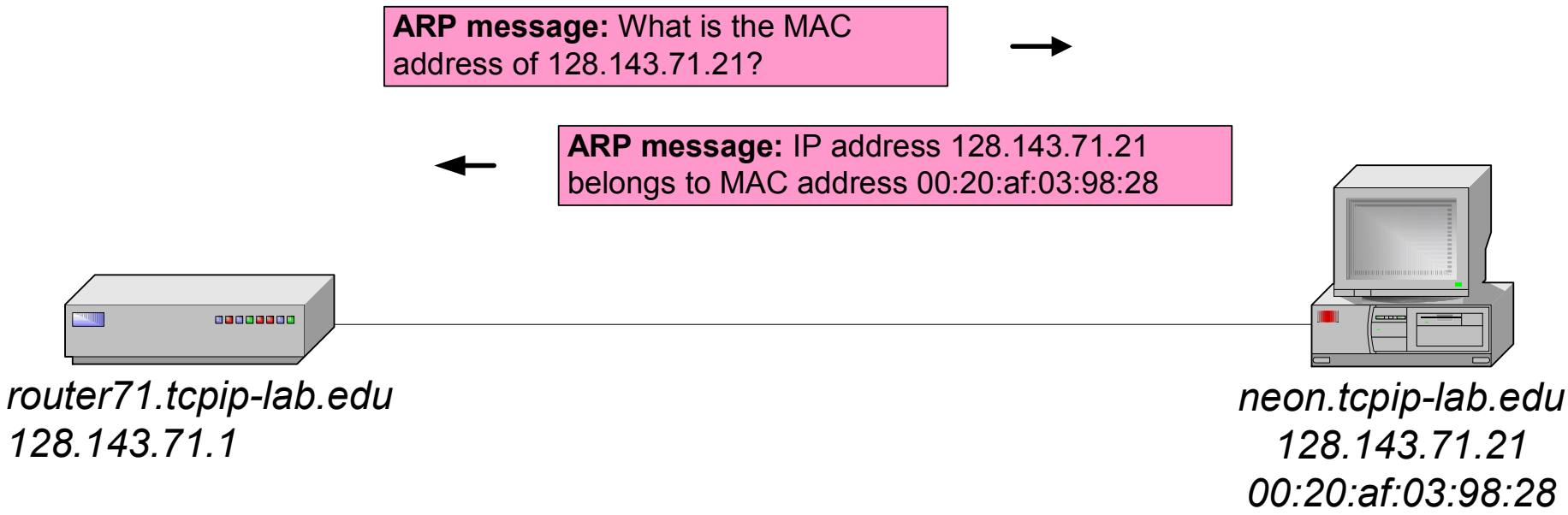
Forwarding the IP datagram

- The IP router receives the Ethernet frame at interface 128.143.137.1, recovers the IP datagram and determines that the IP datagram should be forwarded to the interface with name 128.143.71.1
- The IP router determines that it can deliver the IP datagram directly



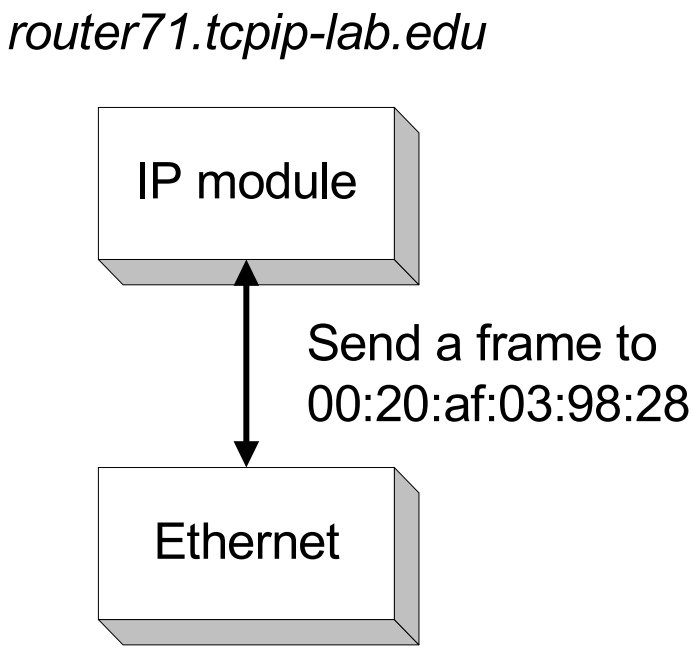
Another lookup of a MAC address

- The router needs to find the MAC address of *Neon*.
- Again, ARP is invoked, to translate the IP address of *Neon* (128.143.71.21) into the MAC address of neon (00:20:af:03:98:28).



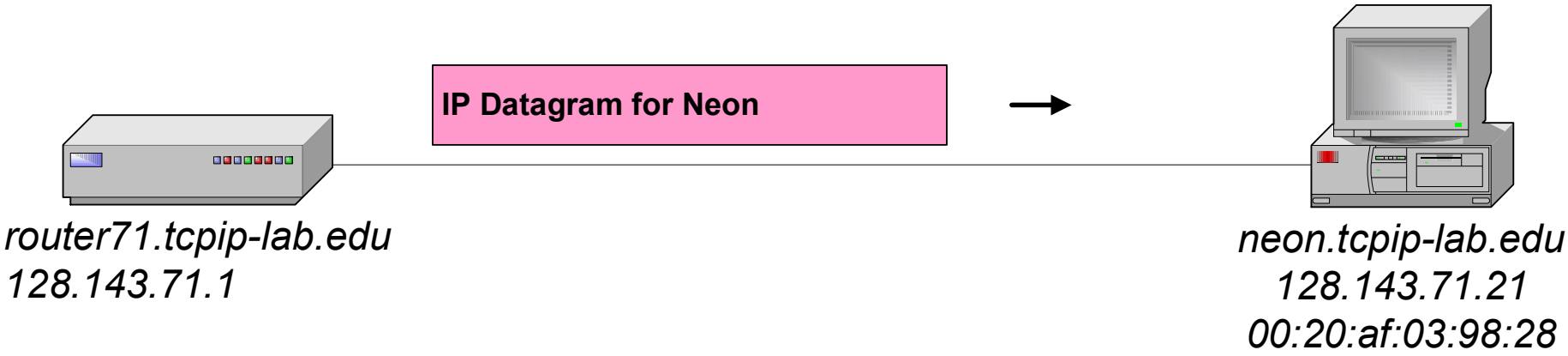
Invoking the device driver at the router

- The IP protocol at *Router71*, tells its Ethernet device driver to send an Ethernet frame to address *00:20:af:03:98:28*



Sending another Ethernet frame

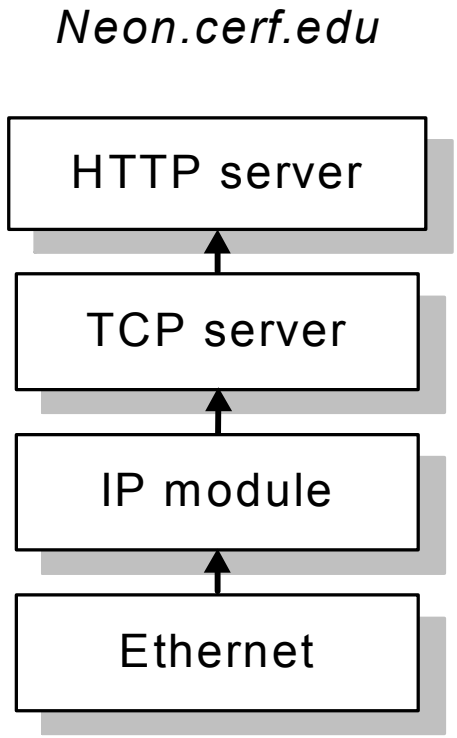
- The Ethernet device driver of *Router71* sends the Ethernet frame to the Ethernet adapter, which transmits the frame onto the wire.



Data has arrived at Neon

- *Neon* receives the Ethernet frame
- The payload of the Ethernet frame is an IP datagram which is passed to the IP protocol.
- The payload of the IP datagram is a TCP segment, which is passed to the TCP server

- **Note**: Since the TCP segment is a connection request (SYN), the TCP protocol does not pass data to the HTTP program for this packet. Instead, the TCP protocol at neon will respond with a SYN segment to *Argon*.



Wrapping-up the example

- So far, *Neon* has only obtained a single packet
- Much more work is required to establish an actual TCP connection and the transfer of the HTTP Request

- The example was simplified in several ways:
  - No transmission errors
  - The route between *Argon* and *Neon* is short (only one IP router)
  - *Argon* knew how to contact the DNS server (without routing or address resolution)
  - ….

How many packets were really sent?

```
tcpdump: listening on fxp0
16:54:51.340712 128.143.137.144.1555 > 128.143.137.11.53: 1+ A? neon.cs. (25)
16:54:51.341749 128.143.137.11.53 > 128.143.137.144.1555: 1 NXDomain* 0/1/0 (98) (DF)
16:54:51.342539 128.143.137.144.1556 > 128.143.137.11.53: 2+ (41)
16:54:51.343436 128.143.137.11.53 > 128.143.137.144.1556: 2 NXDomain* 0/1/0 (109) (DF)
16:54:51.344147 128.143.137.144.1557 > 128.143.137.11.53: 3+ (38)
16:54:51.345220 128.143.137.11.53 > 128.143.137.144.1557: 3* 1/1/2 (122) (DF)

16:54:51.350996 arp who-has 128.143.137.1 tell 128.143.137.144
16:54:51.351614 arp reply 128.143.137.1 is-at 0:e0:f9:23:a8:20

16:54:51.351712 128.143.137.144.1558 > 128.143.71.21.21: S 607568:607568(0) win 8192
                                                          <mss 1460> (DF)
16:54:51.352895 128.143.71.21.80 > 128.143.137.144.1558: S 3964010655:3964010655(0)
                                                          ack 607569 win 17520 <mss
    1460> (DF)
16:54:51.353007 128.143.137.144.1558 > 128.143.71.21.80: . ack 1 win 8760 (DF)
16:54:51.365603 128.143.71.21.80 > 128.143.137.144.1558: P 1:60(59)
                                                          ack 1 win 17520 (DF) [tos
    0x10]
16:54:51.507399 128.143.137.144.1558 > 128.143.71.21.80: . ack 60 win 8701 (DF)
```